ToolBox Mag n° 1

Au sommaire de ce numéro

Editorial — Eric Weyland — Page 3
Numéro 1: c'est parti...

Introduction au C sur GS : première partie — François Uhrich — Page 4
Le début d'une série. Par un expert.

Fontes GS : l'art et la manière — Jean-Yves Bourdin — Page 9
De vraies fontes pour le GS et l'ImageWriter.

Un StartupTool universel pour NDA — Patrick Desnoues — Page 16
Une routine à couper-coller. Un accessoire en prime.

Music Studio Juke-Box — Stephan Hadinger — Page 20
Le premier source d'un Tool GS jamais publié !

Wanted — Jean-Yves Bourdin — Page 25
Idées de programmation : programmez donc utile.

Revue hard : 42 Mégas par disquette ! — Hubert Loiseleux — Page 27
Le disque dur Syquest SQ555 à cartouches amovibles.

GS News — Eric Weyland — Page 29
Les nouvelles vraies du GS : vous pouvez les croire...

Fumées sans feux — Jean-Yves Bourdin — Page 37
Les nouvelles fausses du GS : ne les croyez pas !

Bidouilles, patches et compagnie... — Jean-Yves Bourdin — Page 41
Le patcheur fou a encore frappé.

Courrier des lecteurs — ToolBox — Page 43
Un courrier des lecteurs dans un numéro 1 !

Prix 100 F.

Présentation de la disquette de ToolBox Mag n° 1

Nous ne le répéterons jamais assez, la disquette fait partie de la revue elle-même, elle n'en est pas un complément. Voici donc le catalogue de la partie disquette de la revue :

/ToolBox.MAG.001/DESNOUES.NDA/

La routine de StartupTools et de CloseTools pour les NDAs de Patrick Desnoues. La routine elle-même est contenue dans le fichier SRC InitTools. Le sous-catalogue /LIBRAIRIE/ est là pour vous permettre de réassembler InitTools et le NDA de démonstration 'ToolBox' (il avait bidouillé ses Equates, le bougre...)

Les autres fichiers vous permettront de réassembler le NDA 'ToolBox'. Consultez les premiers fichiers pour voir dans quel ordre ils s'appellent les uns les autres.

A noter : ne lancez pas le fichier de type S16 appelé RESOURCE. C'est la partie Ressources du NDA. Et pensez à mettre le NDA ToolBox dans votre catalogue /SYSTEM/DESK.ACCS/.

/ToolBox.MAG.001/HADINGER.MJUKE/

Le Tool de Jukebox pour les musiques Music Studio de Stephan Hadinger.

MUSICONV : une application pour convertir les musiques Music Studio en musiques Jukebox.

MJEXEMPL : une mini-application pour écouter ces musiques grâce au Tool069. Si vous n'avez pas mis dans /SYSTEM/TOOLS/ le Tool069 de /ToolBox.MAG.001/HADINGER.MJUKE/SOURCES/MJUKE/, ça ne va pas marcher...

ROCK.WBNK, MUS1.MJK, MUS2.MJK, MUS3.MJK : un fichier d'instruments et trois musiques converties à écouter avec MJEXEMPL.

/ToolBox.MAG.001/HADINGER.MJUKE/SOURCES/MUSICONV/ : contient les fichiers sources de l'application MUSICONV.

/ToolBox.MAG.001/HADINGER.MJUKE/SOURCES/MJUKE/ : contient le cœur du programme, le source du fameux Tool069

/ToolBox.MAG.001/HADINGER.MJUKE/SOURCES/EXEMPLE/ : contient le source en Pascal de l'application MJEXEMPL. Quelques lignes de programme, et c'est tout : c'est le Tool069 qui fait tout!

T/ToolBox.MAG.001/HADINGER.MJUKE/SOURCES/INTERFACES/ : les interfaces tout prêts avec le Tool069 pour l'Assembleur, le Pascal et le C : ça, c'est du service.

/ToolBox.MAG.001/HADINGER.MJUKE/SOURCES/ROCK.MUSIQUES/ : 4 musiques Music Studio à convertir et à écouter avec le Jukebox (utilisent les instruments ROCK.WVBNK).

/ToolBox.MAG.001/JYB.BIDOUILLES/

Les bidouilles du patcheur fou. 'TGE.PATCHER' est un fichier EXEC en Applesoft qui sert à patcher The Graphic Exchange pour le faire charger où on veut en mémoire. GRAPHIC.EXCHNGE est un fichier lanceur pour The Graphic Exchange (et tous les gros fichiers SYS) sous GS/OS.

/ToolBox.MAG.001/JYB.FONTES/

Les 'vraies fontes' pour le GS et l'Imagewriter de JYB. Si elles ne vous plaisent pas... changez-les. Bobigny, Meudon : il aurait pu penser à Argenteuil !

/ToolBox.MAG.001/UHRICH.GET/

La commande GET qui manquait à APW, avec son Help. Le source dans un prochain numéro (eh, eh...).

/ToolBox.MAG.001/QUICKIE.DIGITS/

Quelques images numérisées avec le Quickie de Vitesse, en guise de démo de ce scanner qui vous fait tant envie (voir les News d'Eric Weyland).

Le sous-catalogue /DIGITS.320/ contient des images en 16 niveaux de gris très belles sur l'écran du GS. /DIGITS.640/ contient des dessins noir et blanc qui font un excellent 'clip-art'.
En admirant ces images, rappelez-vous : c'est fait en 10 secondes !

APPLE II FORT ET VERT !

C'est parti ! La première revue française dédiée à l'Apple IIGS est entre vos mains.

Son but : le partage des connaissances et du plaisir entre les utilisateurs de GS.

Etant consacrée à l'Apple IIGS, elle se trouve ipso facto totalement indépendante d'Apple : ce paradoxe reflète la situation présente de tous les utilisateurs d'Apple // en France...

C'est la revue de l'Association ToolBox : pas d'autre moyen de se la procurer que d'adhérer à ToolBox en guise d'abonnement (un tarif existera pour les anciens numéros, mais il sera réservé aux membres de l'Association).

Une nouvelle formule de revue

Nous pensons avoir trouvé pour ToolBox Mag une nouvelle formule de revue informatique d'aujourd'hui : la revue avec disquette incluse. Aucune autre revue à notre connaissance ne pratique cette formule : soit la disquette est un supplément facultatif, soit la revue est entièrement sur disquette.

Du coup, vous aurez dans ToolBox Mag des programmes pour GS, mais pas ces pages entières de listings illisibles qu'on trouve ailleurs. Nous mettrons la partie intéressante du source, mais nous ne mettrons pas des centaines de lignes de source assembleur pour décrire... une icône : le source de l'icône sera sur la disquette.

Attention, donc, lecteurs : ToolBox Mag doit se lire GS en route, la disquette et la revue se complètent et renvoient l'une à l'autre.

ToolBox Mag : VOTRE REVUE

Bien entendu, c'est vous qui ferez ToolBox Mag : d'une part en nous disant ce que vous souhaitez y trouver. D'autre part, en y envoyant vos programmes et vos articles. Son sort est entre vos mains.

Un dernier mot : c'est un numéro 1. Soyez indulgents, merci.

Eric Weyland

Imprimé en France – Dépôt légal septembre 1990

Ont collaboré à ce numéro
Jean-Yves Bourdin
Patrick Desnoues
Stephan Hadinger
Hubert Loiseleux
François Uhrich
Eric Weyland

Directeur de publication
Eric Weyland

Rédacteur en Chef
Jean-Yves Bourdin

PAO – Maquettiste
Nathalie Laudat

Secrétariat
Janine Loiseleux

Conseil de Rédaction
Bernard Fournier
Stephan Hadinger
Yvan Koenig
Hubert Loiseleux
Jean-Luc Schmitt
François Uhrich

Siège Social
ToolBox – 6, rue Henri Barbusse
95100 Argenteuil
✆ (1) 30 76 18 64
Télécopie (1) 39 47 44 08

Impression
Imprimerie/Reprographie/Graphisme

ToolBox Mag est un magazine indépendante non rattachée à Apple Computer, Inc. ni à Apple Computer France.
Apple, le logo Apple, Mac et le logo Macintosh sont des marques déposées d'Apple Computer Inc.

—

© ToolBox 1990
Toute reproduction intégrale ou partielle, effectuée par quelque procédé que ce soit, sans l'accord écrit de ToolBox, constitue une contrefaçon. Loi du 11 mars 1957, articles 425 et suivants du Code Pénal.
Droits de traduction, de reproduction et d'adaptation réservés pour tous pays.

C Facile !

François Uhrich

Voici le premier article d'une (je l'espère !) longue série concernant le ténébreux langage C, la mystérieuse ToolBox et l'étrange planète APW/Orca…

Au cours de ces articles, ne comptez pas sur moi pour vous apprendre les bases du langage C : la bible "Programmer en C" de Kerninghan et Ritchie (Editions Masson) est là pour ça !

D'autre part pour pouvoir utiliser la ToolBox du GS dans vos programmes, les Apple IIGS ToolBox Reference volume 1, 2 et tout dernièrement paru le volume 3 (éditions Addison Wesley) sont indispensables.

Même si les exemples sont aujourd'hui un peu dépassés, je vous recommande aussi la Boîte à Outils de l'Apple IIGS par J-P. Curcio (éditions PSI) qui introduit remarquablement à la programmation événementielle sur le GS en langage C.

Du point de vue matériel, le disque dur est, plus encore aujourd'hui, quasiment obligatoire. Le confort d'un environnement "Unixien" tel que APW ne peut être ressenti qu'avec un minimum de changements de disques. D'autre part plus le GS aura de mémoire, plus les répétitives séquences compilation-édition de liens seront rapides…

En bref, un investissement non négligeable aussi bien en matériels, en logiciels ainsi qu'en documentations (par kilos…) est indispensable pour se lancer avec un confort certain dans des développements sérieux en langages évolués sur le GS. Laissons couler une larme furtive au souvenir de nos délires en AppleSoft ou autre Pascal UCSD d'antan…

APW/Orca

Avant de nous lancer dans la comparaison des deux compilateurs C disponibles aujourd'hui sur le GS, voyons tout d'abord l'environnement de développement Apw/Orca.

Pourquoi cette association ? Tout simplement parce que ces deux produits n'en font qu'un! A l'origine APW a été développé par ByteWorks. A l'heure actuelle, le Shell APW (produit Apple) en est à la version 1.02 et le Shell ORCA/M (produit ByteWorks) à 1.1 en attendant une future version 2.0 de ByteWorks qui réunira les 2 branches distinctes.

Quelles différences entre ces deux produits ? Minimes.

En surface, ORCA/M 1.1 possède deux commandes Shell internes supplémentaires et non fondamentales :

Compress : alphabétisation et compression des répertoires.

Switch : échange de position entre deux fichiers dans un répertoire.

Plus importante est une optimisation de l'utilisation mémoire par ORCA/M pour conserver en RAM les fichiers couramment utilisés. Ainsi, lors d'une compilation suivie immédiatement par l'édition de liens (commandes CMPL ou RUN), le ou les fichiers objets sont conservés en RAM et peuvent même ne pas être sauvés sur disque (option +M). L'édition de liens s'en trouve considérablement accélérée.

Ce qui fait la force de cet environnement par rapport à ses concurrents (je pense à Merlin16 pour l'assembleur et à TML Pascal 2), c'est qu'il offre une plate-forme de développement commune et cohérente pour de nombreux langages qui peuvent être utilisés simultanément dans un même projet. A l'heure actuelle les langages suivants sont disponibles :

– ASM65816 : assembleur commun à Apple et Byte Works sauf qu'il existe des versions de macros légèrement différentes suivant la provenance.

– TML Pascal APW : ce compilateur n'a pas été mis à jour pour le système 5.0 et n'est donc pas encore compatible avec TML Pascal II.

– Orca/Pascal : continuellement mis à jour par ByteWorks comme tous leurs produits.

– Orca/C.

– APWC.

– REZ : langage descriptif des ressources du système 5.0. Inclus dans le package APW Tools and Interfaces v1.1 avec tous les utilitaires nécessaires à la manipulation des ressources, un autre éditeur de liens LinkIIGS ainsi que les interfaces C et Assembleur du système 5.0.

– EXEC : langage interprété composé de commandes du Shell. Parent lointain des Shell Scripts d'UNIX.

– LINKED : langage du Linker v1.0 tombé en désuétude.

– LINKER : langage des scripts du nouvel éditeur de liens de Byte Works : Linker v1.2 alias ZapLink.

– ProDOS et TEXT.

Tous les compilateurs pour ces langages sont regroupés dans le répertoire du préfixe 5/ normalement intitulé LANGUAGES.

Les librairies associées sont quant à elles stockées dans le répertoire du préfixe `2/` (`LIBRARIES`) avec également toutes les macros et equates Assembleur (`2/AInclude`), les includes `C` (`2/CInclude` ou `2/OrcaCDefs`) et les units Pascal (`2/OrcaPascalDefs`).

Le préfixe `3/` désigne le répertoire des fichiers temporaires du système tel que `SYSTEMP` le presse-papiers de l'éditeur texte.

Le préfixe `4/` (`SYSTEM`) désigne le répertoire des fichiers systèmes de APW : notamment l'éditeur texte, le fichier SYSCMND décrivant toutes les commandes internes et externes (préfixe `6/`) du Shell et le fichier LOGIN en langage `EXEC` toujours exécuté à l'arrivée dans APW. A noter que puisque la recherche de ce LOGIN a lieu avant la moindre possibilité de modification des préfixes, il faut qu'il y ait un répertoire `SYSTEM` au même niveau que le Shell `ORCA.SYS16` ou `APW.SYS16` qui a été lancé. Ce répertoire n'a pas à contenir tout un système GS/OS s'il ne s'agit pas du volume de démarrage mais il doit s'appeler SYSTEM.

Enfin l'important préfixe `6/` (`UTILITIES`) désigne le répertoire contenant tous les utilitaires de APW. Ecrits en langage EXEC ou programmes de type EXE, tous ces utilitaires consacrent la puissance de APW. L'écriture en langage C de ces utilitaires sera abordée prochainement.

APWC

Honneur au premier langage évolué natif disponible sur Apple IIGS dès sa sortie.

APWC en est aujourd'hui à la version 1.02. Il peut être acheté séparément ou avec l'environnement APW. Le package comprend une seule disquette 3,5" et l'installation dans l'environnement APW est aisée.

La disquette inclut 3 exemples intéressants mais qui pour l'un d'entre eux montrent une faiblesse de APWC : l'écriture de NDA est possible mais à l'aide d'une cuisine pour le moins indigeste et surtout nécessitant un code par trop important.

La documentation technique comprend 6 chapitres et 6 appendices. Sans aucune pédagogie, elle fera fuir les débutants. La moitié des 170 pages, qui la composent, est consacrée à une description sèche de la librairie.

Le gros défaut de APWC est la taille du code qu'il génère. En effet APWC réserve sa page directe et son espace mémoire pour sa pile au moment de l'édition de liens (link) au lieu de le faire par des appels au Memory Manager lors de l'exécution. Ainsi le programme vide main { } à l'issue de la

Orca/C et APWC ensemble

Les deux langages C peuvent tout à fait être présents en même temps dans l'environnement. Le problème est que les 2 compilateurs ont le même nom : CC.

Il faut par exemple renommer le compilateur Apple APWC avant de le copier dans le répertoire des langages 5/. De plus, il faut modifier le fichier 4/SYSCMND en conséquence :

```
CC      *L   8     ORCA/C
APWC    *L   10    APW C compiler
```

Les nombres 8 et 10 sont utilisés dans le type auxiliaire des fichiers sources pour les différencier : ainsi #CHANGE MonSource CC revient à mettre la valeur 8 comme type auxiliaire du fichier MonSource. Lorsqu'une compilation est demandée (commandes compile, assemble, cmpl, run), c'est ce numéro qui permet au Shell de savoir quel compilateur il doit lancer.

Depuis la mise à jour de Orca/C v1.1 B11, on peut arriver à conserver les librairies des deux langages C dans le même répertoire 2/. Toutefois certaines priorités dans l'ordre doivent être respectées : ainsi ORCALIB (ou ORCAGLIB) doit être en tête, suivi par PASLIB si Orca/Pascal est présent, puis SYSLIB et enfin CLIB. Toutefois avec APWC, il faut alors indiquer au moment du link la librairie à ouvrir en priorité :

#CMPL MonSource.cc 2/CLib keep=MonProg
ou bien après la compilation :
#LINK 2/Start MonSource 2/CLib keep=MonProg

Le Portage d'un compilateur à l'autre est assez facile. A remarquer toutefois dans l'utilisation de la ToolBox, que le code d'erreur renvoyé par la fonction est stocké en APWC dans la variable _toolErr alors qu'avec Orca/C il faut appelé la fonction toolerror(). Il n'y a pas de PRAGMA en APWC. Enfin la segmentation est légèrement différente et cela à un point-virgule près :

APWC : segment "*MonSegment*"
Orca/C : segment "*MonSegment*";

compilation et du link a pour taille plus de 8000 octets (segment ExpressLoad inclus) ! Pourtant un avantage non négligeable de ce code est une très grande fiabilité, autant de la part du compilateur que de la librairie C.

Toutefois un problème est que la dernière mise à jour du compilateur et de la librairie remonte à septembre 1988. Ainsi les accès disque effectués par cette librairie n'utilisent pas les appels spécifiques GS/OS mais toujours ceux de ProDOS 16... Cela n'empêche pas bien sûr les programmes de tourner sous GS/OS, mais il y aurait moyen de gagner du temps et du code.

La solution peut être de ne pas utiliser la librairie pour les accès fichiers et d'appeler GS/OS directement : bref, de se faire sa propre librairie...

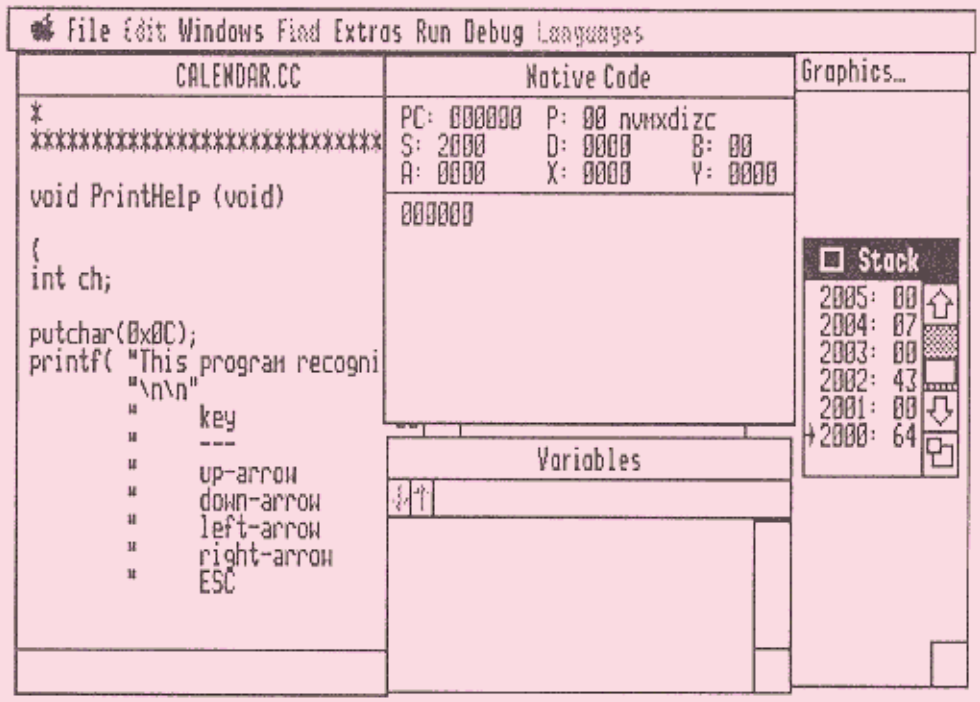
Orca/Desktop ou Prizm est avant tout un éditeur graphique excellent et extrêmement rapide. Il permet de visualiser plusieurs sources simultanément grâce au multi-fenêtrage.

Une fenêtre peut même être divisée en 2 parties quasi indépendantes. D'autre part Prizm permet de compiler, de linker et possède même, fait unique dans l'histoire de l'Apple II, un debugger symbolique (Pascal et C) intégré !

A l'usage Prizm se révèle être très utile en tant qu'éditeur mais pas toujours fiable, dans sa version actuelle 1.0b, en tant qu'environnement de développement, dès que l'application devient un tant soit peu importante. Ainsi personnellement, j'ai ajouté Orca/C à APW/ORCA, Prizm est un utilitaire en préfixe 6/. Je l'utilise pour éditer mes sources puis je quitte Prizm pour compiler sous APW/Orca.

Les interfaces quant à elles ont été mises à jour dans le package APW Tools and Interfaces v1.1. On peut donc parfaitement développer des applications en APWC utilisant les dernières nouveautés de la ToolBox y compris les ressources.

En bref, APWC reste une bonne voiture de tourisme fiable qui, malgré son âge, tient encore la route notamment pour de gros projets (une page de pub : Tarot a été développé entièrement en APWC...), mais qui aurait besoin d'une petite révision. Il n'est pas à recommander aux débutants.



Les fenêtres de Prizm

ORCA/C

Orca/C a été édité en juin 1989 par ByteWorks. Une nouvelle version du compilateur (v1.1 B11) de la librairie et de certains includes ont été envoyés gratuitement en juin 1990 à tous les acheteurs enregistrés.

L'ensemble comprend trois disquettes et un imposant et remarquable manuel. Le langage peut être installé dans l'environnement APW/ORCA ou bien être utilisé indépendamment grâce à Orca/Desktop (alias Prizm) inclus sur l'une des 3 disquettes. Une de ces disquettes est consacrée exclusivement à des exemples variés et instructifs. A noter la présence de macros Assembleur offrant des facilités étonnantes pour interfacer du C avec des routines ASM65816.

La compilation est un peu plus rapide avec Orca/C qu'avec APWC mais c'est surtout au link que APWC est très nettement dépassé.

Le compilateur Orca/C accepte de nombreuses directives importantes sous forme de pragmas dans le source. Il est notamment très facile de créer un NDA ou un CDA :
#PRAGMA NDA open close action init period eventmask menuLine.
#PRAGMA CDA name start shutdown.

Ces Pragmas indiquent au compilateur qu'il lui faut créer l'en-tête correspondant aux Accessoires de Bureau et non pas celui d'un programme classique.

Suivant la taille finale de votre programme, il vous est possible de spécifier au compilateur le modèle de mémoire à utiliser :

- Code et Data sur moins de 64Ko (#PRAGMA MemoryModel 0). Cela permet au compilateur de générer du code avec un adressage court (adresse sur 2 octets) plus rapide et plus compact.

- Code et Data sur plus de 64Ko (#PRAGMA MemoryModel 1). L'adressage dans ce cas sera long (sur 3 octets). Une version de la librairie correspond à chacun des 2 modèles.

Toujours sous forme de Pragma, il est possible d'indiquer l'utilisation de la carte coprocesseur mathématique FPE d'Innovative Systems et le compilateur générera le code utilisant directement cette carte ! C'est actuellement le seul compilateur à offrir cette possibilité. Une autre option est la génération de code pour le debugging : à tel point qu'il est possible durant l'exécution de faire apparaître, avec toutefois certaines réserves, le numéro de la ligne où une erreur a été détectée !

Enfin lorsque la version définitive de votre sublime logiciel est achevée, il vous est possible d'indiquer au compilateur de générer du code optimisé par plusieurs méthodes simultanément ! Attention toutefois, car si vous n'avez pas été suffisamment rigoureux dans le passage des paramètres aux fonctions, il peut arriver que tout fonctionne à première vue parfaitement dans la version non optimisée et que tout se gâte après l'optimisation...

Le code généré est très nettement optimisé par rapport à APWC. Le programme vide main(){} ne nécessite plus que 950 octets (segment ExpressLoad inclus).

371 pages composent l'excellente documentation. Celle-ci inclut une remarquable introduction à la programmation "desktop" du GS ainsi qu'un manuel complet de référence du langage C. Le tout est saupoudré de nombreux exemples. L'installation et l'utilisation sous Orca/Desktop ou bien sous Shell APW/ORCA sont toutes deux traitées.

Alors que l'utilisation de APWC ne nécessite pas de licence particulière, il n'en est pas de même avec Orca/C. Si vous diffusez sous quelque forme que ce soit (FreeWare, ShareWare ou autre) un programme utilisant ne serait-ce qu'une seule fonction de la librairie Orca/C, vous devez inclure dans la documentation et le programme un texte standard indiquant que ce logiciel a été écrit avec Orca/C de ByteWorks Inc. Vous devez également envoyer un exemplaire de votre application à ByteWorks accompagné de votre signature sur un accord type joint au manuel. En fait ByteWorks considère que la publicité ainsi faite à ses produits couvre les frais d'utilisation de la librairie C : il n'y a donc rien à payer.

La première version de la librairie n'était pas, et de loin, sans problème. Le développement de programmes était sans cesse freiné par des réactions bizarres de la librairie. Les choses semblent s'être améliorées depuis, avec la mise à jour de juin dernier, mais prudence...

En conclusion Orca/C est une belle formule 1 sans doute plus difficile à régler que APWC mais qui est toutefois à recommander aux débutants pour son magnifique manuel. De plus un réel suivi par ByteWorks existe, alors que APWC ne semble plus évoluer.

NDLR : il existe un troisième environnement de développement pour programmer en C pour le GS : MPW-C/GS pour le Mac. De l'avis unanime des développeurs

Orca/C et le système 5.0

Plusieurs personnes m'ont contacté pour me demander comment j'avais pu utiliser les ressources et autres merveilles du système 5.0 dans mes programmes en Orca/C (notamment TransProg) ?

Rien de plus simple.

Depuis le système 4.0, les interfaces ToolBox des langages ont été standardisées par Apple. Ainsi les includes pour APWC que l'on peut trouver dans APW Tools and Interfaces v1.1 sont tout à fait utilisables avec Orca/C.

Pour ce faire, il suffit de copier de /APW.Interfaces/CInclude tous les includes concernant la ToolBox et non la librairie propre au compilateur dans le répertoire 2/ORCACDEFS des includes Orca/C : copier donc `ACE.H`, `ADB.H`, `APPLESHAREFST.H`, `CONTROL.H`, `DESK.H`, `DIALOG.H`, `EVENT.H`, `FONT.H`, `GSOS.H`, `INTMATH.H`, `LINEEDIT.H`, `LIST.H`, `LOADER.H`, `LOCATOR.H`, `MEMORY.H`, `MENU.H`, `MIDI.H`, `MISCTOOL.H`, `NOTESEQ.H`, `NOTESYN.H`, `PRINT.H`, `PRODOS.H`, `QDAUX.H`, `QUICKDRAW.H`, `RESOURCES.H`, `SANE.H`, `SCHEDULER.H`, `SCRAP.H`, `SOUND.H`, `STDFILE.H`, `TEXTEDIT.H`, `TEXTTOOL.H`, `VIDEO.H`, `WINDOW.H` et également `TYPES.H`.

Puis il suffit de changer le langage (c'est-à-dire le type auxiliaire) de tous ces fichiers de APWC en ORCA/C. Pour cela il faut utiliser la commande Shell CHANGE :

CHANGE 2/ORCACDEFS/= CC

Tous (le signe =) les fichiers sources de ce répertoire seront alors dans le langage Orca/C.

consultés, il est affecté des mêmes défauts qu'APW-C, essentiellement l'encombrement du code-objet en espace-disque et en espace-mémoire. Ses librairies seraient entièrement à refaire.

C'est au point que Sierra-On-Line, éditeur de jeux d'aventures animés (série des King Quest etc), a annoncé récemment qu'il ne produirait plus de versions GS de ses jeux tant qu'il ne disposerait pas d'un bon compilateur C.

Documentations

- Programmer en C de Kerninghan et Ritchie (Masson) 23.
- Apple IIGS ToolBox Reference Vol 1, 2 et 3 (Addison Wesley) 123.
- Boîte à outils de l'Apple IIGS de J.P. Curcio (PSI) 23.

Logiciels

APW : Apple IIGS Programmer's Workshop v. 1.0.2 *(I)*

APW C : Apple IIGS Programmer's Workshop C v. 1.0.2 *(I)*

APW Tools and Interfaces v.1.1 *(I)*

ORCA/M 1.1 *(IV & V)*

ORCA/C 1.1 B11 *(IV & V)*

Tarot *(V)*

I – **Prim'Vert Editions** (APDA France)
36, rue des Etats Généraux – 78000 Versailles
✆ *(1) 39 02 33 44*

II – **Inter Editions**
25, rue Leblanc – 75842 Paris Cedex
✆ *(1) 40 60 40 64*

III – **Infothèque**
32, rue de Moscou – 75008 Paris
✆ *(1) 45 22 67 01*

IV – **ByteWorks Inc.**
4700 Irving Bld. NW Suite 207
Albuquerque, NM 87114 – USA

V – **ToolBox**
6, rue Henri Barbusse – 95100 Argenteuil
✆ *(1) 30 76 18 64*

Le cadeau ToolBox : une commande 'GET' pour APW

GET est la commande externe APW/ORCA qui manquait cruellement au Shell !

En effet, il était impossible jusqu'à présent à l'utilisateur d'APW d'intervenir dans l'éxécution d'un programme EXEC par l'intermédiaire d'une saisie au clavier.

GET est écrit en ORCA/C et occupe moins de 2Ko avec ExpressLoad... Le source commenté vous sera présenté prochainement.

Syntaxe de GET

GET variable [texte]

Le texte s'il est présent est affiché sur la sortie standard, puis l'utilisateur doit frapper une touche qui deviendra la nouvelle valeur de la variable indiquée.

Paramètres :

variable	*variable qui sera affectée.*
texte	*texte éventuel qui sera affiché. Si le texte contient plusieurs mots, il doit être encadré de guillemets.*

Exemple :

GET Réponse "Votre Choix ? "

Installation

Copier le fichier GET de type EXE dans votre répertoire des utilitaires (6/). Copier l'aide GET du dossier HELP dans 6/HELP. Editer le fichier 4/SYSCMND et ajouter la ligne suivante :
GET *U Saisie Touche Clavier par François Uhrich août 1990

L'étoile devant le U, pour utilitaire, indique que GET peut rester résident en mémoire.

Exemple de LOGIN

Le fichier 4/LOGIN est automatiquement exécuté dès l'arrivée dans APW/ORCA. GET peut par exemple vous permettre de choisir par menu le répertoire du source sur lequel vous voulez travailler ou bien de lancer PRIZM... etc.

```
show time                          alias ed      edit
* pour les UNIXiens...             echo
alias cd      prefix               get key "(U)ser (P)rizm "
alias cp      copy                 if {key} == "P"
alias del     delete               prizm
alias rm      delete               else if {key} == "U"
alias ls      cat                  prefix 6/../user
alias mv      move                 end
alias ren     rename               echo
alias mkdir   create               unset key.
```

Fontes GS : savoir s'en servir

Jean-Yves Bourdin

Un mythe à dissiper

Pour savoir vraiment se servir des fontes GS, il faut tout d'abord dissiper un mythe : l'idée qu'il n'existerait qu'une sorte de fontes GS. Si vous croyez cela, vous mettrez toutes vos fontes dans le même sac, et vous n'en sortirez jamais.

Ce mythe est vrai, bien sûr, sur un point décisif : ce sont toutes des fontes QuickDraw, gérées par la Rom du GS et du Mac, donc éditables avec votre éditeur de fontes. En conséquence, le présent article concerne aussi bien les utilisateurs de TimeOut SuperFonts que ceux de Publish-It ou que les utilisateurs de Mac, puisque ce sont les mêmes fontes.

Mais cela s'arrête là : les fontes Apple ont en effet des choses fort différentes à faire, et les mettre toutes dans le même sac ne fait qu'engendrer la confusion.

Cette confusion a son origine dans l'Apple lui-même : Apple ayant voulu dès le départ que le Mac et le GS travaillent en "Tel écran Tel écrit", c'est volontairement qu'il a fait une seule sorte de fontes. L'écran de l'Apple est censé être la représentation fidèle de la feuille de papier.

Seulement, un écran GS, un écran Mac, une ImageWriter, une LaserWriter, ce n'est pas du tout la même chose. Ce n'est en particulier pas du tout la même résolution (le nombre de points par centimètre-carré). Dès qu'on utilise d'un peu près les fontes GS, on s'aperçoit qu'il faut en fait subdiviser, et se constituer des types de fontes adaptées à leur périphérique de sortie.

Car si on se sert des mêmes fontes pour tout, ça marche, puisque le GS est prévu pour ça. Seulement, vous êtes dans la situation de celui qui mettrait les mêmes gants pour conduire sa voiture, pour sortir en ville, pour faire du ski, et pour boxer, sous prétexte que ce sont toujours des gants : ça ne marche vraiment bien que dans l'une des fonctions, dans les autres le résultat varie de cocasse à lamentable.

Un petit mot pour ceux –j'en suis– qui trouvent que l'impression en mode graphique est bien lente. Elle l'est, effectivement. Et sur toutes les machines. C'est le prix de la qualité de l'impression : plus on veut de résolution, plus on fait de calculs, et plus on demande à l'imprimante.

Ne comptez pas sur les buffers d'imprimante, ils ne servent à rien en mode graphique. D'une part, il y a beaucoup trop d'octets à envoyer, et d'autre part le GS communique avec l'imprimante dans les deux sens, il a donc besoin de recevoir des messages de l'imprimante.

Pour accélérer l'impression, je ne vois que la Transwarp et l'augmentation de la Ram ; la page est dessinée en mémoire avant d'être imprimée.

En-dehors de cela, la solution, sur GS comme sur Mac c'est le spooling, ou l'impression en tâche de fond : l'ordinateur imprime encore plus lentement, mais il le fait dans les périodes où vous le laissez sans rien faire. Le problème n'est en effet pas tant la lenteur de l'impression en elle-même, que l'indisponibilité de l'ordinateur pendant ce temps. Je ne connais pas encore d'application qui fasse cela sur GS, mais ça ne saurait tarder, car c'est dans la Rom du GS, le Print Manager : voyez l'article "Wanted" de ce numéro.

Les cinq genres de fontes

Je distinguerai donc au moins les types suivants :

Les fontes pour l'écran du Mac

Paresseux comme toujours, Apple met dans le sous-catalogue */FONTS* des disquettes système GS des fontes (*Geneva*, *Chicago*) reprises directement des disquettes système du Mac. Si ces fontes sont sur les systèmes Mac, c'est qu'elles sont bien adaptées à l'affichage sur l'écran du Mac.

Seulement voilà : alors que les points de l'écran du Mac (pixels) sont carrés, ceux du GS sont rectangulaires (deux fois plus hauts que larges).

Se servir de ces fontes pour afficher sur l'écran du GS fait donc ces lettres allongées que nous voyons si souvent sur nos écrans. Sur ImageWriter, leur résultat est plutôt laid. Sur LaserWriter, le driver leur substitue à l'impression les fontes internes de la Laser : adieu au "Tel écran Tel écrit".

Bref, sur GS, ces fontes sont parfaitement inutiles. Nous pouvons les jeter. Sauf dans un cas : le cas suivant.

Les fontes pour l'écran 320 du GS

Le GS a, en effet, deux modes différents d'affichage sur son écran : mode 640 points de large (le mode habituel des applications), et le mode 320. En mode 320 points, il affiche les fontes en mettant deux rectangles (pixels) l'un à côté de l'autre au lieu d'un seul.

C'est là qu'est le truc : en mettant l'un à côté de l'autre deux rectangles deux fois plus hauts que large, vous obtenez un carré. Les fontes "écran Mac" sont alors réutilisables (si elles sont bien dessinées, parce qu'elles sont grossies). Mais attention : seulement pour vos affichages en mode 320, dans vos programmes, ou quand vous mettez du texte dans un dessin en mode 320 avec votre programme de Paint ou de Draw. Bref, ces fontes ne doivent servir qu'à l'affichage sur l'écran 320, et pas à imprimer, par exemple.

Pour voir ce qu'une fonte GS donne en mode 320, utilisez la fonction "Afficher une fonte" de Fontasm 2.0, et pas la fonction "Charger une fonte". Vous aurez alors dans le menu Pomme la possibilité d'afficher cette fonte en mode 320.

Les fontes pour l'écran 640 du GS

En mode 640, le GS a besoin, pour afficher proprement sur son écran, de fontes qui tiennent compte de ce que ses points sont rectangulaires, et ne fassent pas d'affichage "allongé". Il y a effectivement une fonte Apple qui est dans ce cas. C'est la fonte *Shaston* (*Shaston 8* dans la Rom, *Shaston 16* sur les disques systèmes). C'est donc elle qui produit les affichages les plus lisibles sur l'écran du GS.

C'est d'ailleurs pourquoi les utilisateurs de traitement de textes sur GS doivent séparer les fonctions de saisie du texte et la fonction de "mise en fontes" pour l'impression. Pendant la saisie du texte, le mieux est de choisir la fonte *Shaston 8*, la plus lisible. Ensuite, on change les fontes avant d'imprimer.

En dehors de *Shaston*, il y a fort peu de fontes expressément conçues pour l'affichage sur l'écran 640 du GS. J'en ai cependant trouvé quelques-unes qui, par hasard, conviennent.

Tout d'abord, certaines fontes Mac qui étaient destinées à produire un affichage "compressé" sur le Mac : les fontes *Stuttgart*, par exemple, donnent sur GS un affichage normal. Il y en a d'autres.

Ensuite, les fontes de... Publish-It ! Attention, cependant : Publish-It paye la lisibilité de ses fontes sur l'écran Apple // d'une qualité moindre à l'impression sur ImageWriter (voir ci-contre). Je ne recommande pas de s'en servir pour imprimer.

Les fontes correspondant à celles de la LaserWriter

Là, les choses sont simples : les fontes qui sont dans votre dossier `/SYSTEM/FONTS` sont la représentation sur écran des fontes internes de l'imprimante. Elles n'en sont qu'une représentation imparfaite : ne vous souciez pas de leur aspect, le driver de LaserWriter leur substituera tout seul les fontes de l'imprimante.

Enfin, il le devrait du moins : pour certains caractères spéciaux qui sont dans les fontes-écran mais pas dans celles de la Laser, le driver de Laser devrait appeler automatiquement les caractères correspondants de la fonte *Symbols*. Il le fait sur Mac, mais sur GS il oublie : c'est un bug. Il faudra donc le dire "à la main" à vos programmes.

Si vous êtes l'heureux utilisateur d'une LaserWriter, nous vous conseillons donc de ne pas toucher du tout à ces fontes-écran, sauf pour ces caractères spéciaux où nous vous recommandons de mettre un petit pense-bête pour appeler *Symbols*.

Bien entendu, vous avez le droit d'imprimer sur Laser directement avec des fontes écran : simplement, ne le faîtes pas, c'est vraiment moche. Les fontes PostScript de l'imprimante sont d'une qualité incomparable.

Au fait, les avez-vous, ces fontes ? Parce que, quand vous achetez une LaserWriter à un concessionnaire Apple en France pour la connecter à votre GS, ces fontes ne vous sont fournies que sur une disquette Mac. Il n'y a rien pour le GS, ni fontes-écran, ni programme de chargement de fontes Postscript dans l'imprimante, ni programme d'envoi d'un texte en PostScript. Bravo à l'Apple // Service Team d'Apple France, ça c'est du service !

Heureusement, il y a ToolBox. Si vous avez un GS et une LaserWriter, contactez ToolBox, nous avons traduit les fontes de la NT pour le GS (*Avant-Garde*, *Bookman*, *Courier*, *Helvetica*, *Helvetica Narrow*, *New Century Schoolbook*, *Palatino*, *Symbols*, *Times*, *Zapf Chancery*, *Zapf Dingbats*).

Les fontes pour l'ImageWriter

Dans le fond, c'est de cette cinquième sorte de fonte dont nous avons tous besoin : des fontes prévues pour l'impression en qualité maximale sur ImageWriter à partir du GS. Eh bien, il m'a fallu pas mal de temps pour le découvrir : ces fontes là n'existent pas !

Elles n'existent pas, mais nous avons moyen de nous les fabriquer. C'est l'objet de la deuxième partie de cet article.

De vraies fontes pour l'ImageWriter

❄ Les fontes allongées

Réglons d'abord une question simple : celle des fontes qui apparaissent allongées, étirées, sur l'écran et le papier. La raison, déjà étudiée (en page 10) : ce sont des fontes issues du Mac, qui a des pixels carrés, alors que les pixels du GS sont rectangulaires. Il y a deux solutions simples :

— soit vous utilisez des fontes qui n'apparaissent pas étirées, comme les fontes Stuttgart ou les fontes de Publish-It, et le problème s'arrête là ;

— soit vous utilisez les fontes Mac habituelles, en acceptant qu'elles apparaissent étirées sur l'écran. Dans ce cas, le driver d'ImageWriter a une option pour rétablir les bonnes dimensions à l'impression : il s'agit de l'option «Condensé» du menu «Page Setup» ou «Mise en Pages».

Nous conseillons aux utilisateurs d'Imagewriter d'utiliser cette option : cela augmente en fait la résolution, donc la qualité, de l'impression. Si vous tenez à des proportions normales sur l'écran, utilisez Graphic Writer III, de Seven Hills : il a une option pour présenter ces fontes normalement.

A partir de maintenant, nous admettrons que vous utilisez ces fontes "allongées", avec l'option "Condensé", pour obtenir une qualité maximale d'impression.

❄ Des dessins allongés

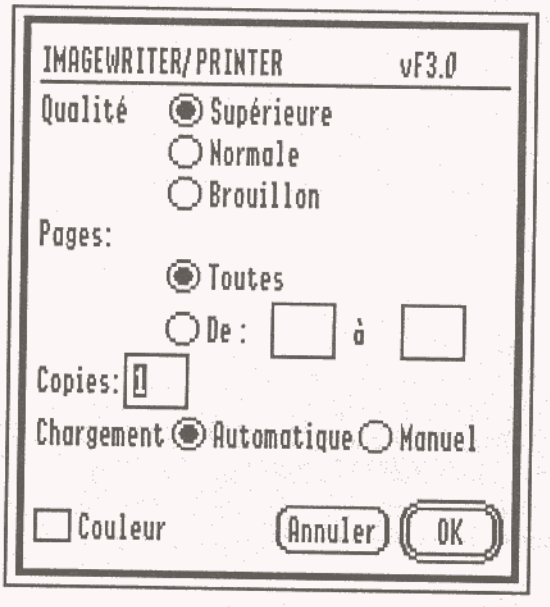
Il y a une conséquence un peu inattendue de l'option "Condensé" : elle condense aussi les dessins ! Un dessin GS "normal" au milieu d'un texte imprimé avec BeagleWrite GS ou AppleWorks-GS verra sa hauteur diminuée par deux. Il faut donc, si vous imprimez en condensé, doubler la hauteur de vos dessins.

Graphic Writer III se charge automatiquement de doubler la hauteur des dessins quand il les "importe". Avec AppleWorks-GS, il faut doubler soi-même, à la main, ces dessins dans la partie graphique d'AppleWorks-GS. Sinon, on peut doubler auparavant la hauteur des dessins avec PaintWorks-Gold ou tout autre programme de Paint.

❄ Des fontes par couples

Même avec l'option "Condensé", il reste vrai que la résolution de l'imprimante est bien supérieure à celle de l'écran. Il reste vrai aussi que la qualité de l'impression n'est correcte que si vous utilisez la résolution maximale de l'imprimante.

Apple s'en est sorti par un "truc" qui n'est pas clairement expliqué dans ses documentations. Quand vous choisissez la qualité "Supérieure" ou "Better Text" dans le menu ⌘-P, le GS va chercher la fonte de taille double de celle demandée (New York 24, par exemple, si vous voulez imprimer en New York 12), et imprimera en réduisant de moitié cette fonte double, tout en gardant la résolution de la fonte double. Le résultat sur papier est d'excellent niveau. *Notons que TimeOut SuperFonts pour AppleWorks fait la même chose.*



L'utilisateur doit prendre garde : si vous voulez imprimer correctement avec *TIMES.12*, par exemple, il faut avoir *TIMES.24* dans votre catalogue `/SYSTEM/FONTS`, et choisir «Better Text» ou "Supérieure" dans le menu d'impression. Si vous n'avez que *TIMES.24*, ça ira : votre écran affichera une

bouillie désagréable, mais l'impression papier sera impeccable. Si vous n'avez que *TIMES.12*, ça n'ira pas : l'écran sera convenable, mais sur papier, l'impression sera affreuse. Bref, il faut diviser par 2, mentalement, la taille de vos fontes. Si vous avez une fonte *36*, son impression ne sera correcte que si vous demandez du *18* dans la même fonte.

Pour le possesseur d'Imagewriter, cela signifie la règle d'or suivante : il doit organiser ses fontes par couple. La fonte de taille *24* sera destinée à l'impression, la fonte de taille *12* à la représentation sur écran.

Attention : les logiciels vont vous proposer d'afficher et d'imprimer en taille 24. Refusez, sauf si vous avez la fonte de taille 48 dans votre système ! Refusez également leur proposition d'imprimer dans des tailles dont vous n'avez ni la fonte simple, ni la fonte double. Si vous demandez une taille 14 pour une fonte où vous n'avez que la taille 12 dans votre système, QuickDraw va se "débrouiller", sur l'écran comme sur le papier, pour reconstituer une 14 à partir de la 12. Mais ça ne sera vraiment pas beau !

❄ *De vrais couples*

Les choses seraient finalement assez simples si quelqu'un avait fabriqué pour le GS des fontes spécifiques par couple : Apple, par exemple, pourrait bien nous mettre sur les disques système des fontes correspondant aux spécifications de son driver d'imprimante. Beagle, avec les fontes de SuperFonts ou les fontes ex-Styleware, aurait dû nous vendre ce genre de fontes.

Ca marcherait sans problème si les couples de fontes Mac étaient de vrais couples : si, par exemple, chaque caractère de la fonte *GENEVA.24* était exactement le double, en hauteur et en largeur, de la fonte *GENEVA.12*. Malheureusement, ce n'est **strictement jamais** le cas !

Même si on imprime avec la fonte double, il faut bien quand même que l'écran serve à quelque chose, à savoir la mise en pages. Si la fonte simple n'est pas la réduction exacte à 50% de la fonte double, l'écran nous trompe : justifications ratées, lignes décalées, etc. La version 1.1 d'AppleWorks-GS fait ce qu'elle peut pour s'adapter à cette situation. Mais il reste que l'écran ne permet pas réellement de gouverner l'impression si les fontes ne sont pas strictement proportionnelles.

D'autant que, si vous prenez une fonte *24* pour l'imprimer en *12*, vous aurez de toute façon des surprises : la fonte n'a pas été conçue pour être imprimée en réduction. Un point blanc entre deux points noirs est visible en *24*, mais disparaît pratiquement si on imprime en réduction. Une distance d'un seul point pour le trait de soulignement est visible sur l'écran, mais disparaît pratiquement également à l'impression.

Vous comprenez maintenant pourquoi j'ai écrit plus haut que les fontes GS pour l'Imagewriter n'existaient pas : les **vraies** fontes GS correspondant aux spécifications du driver d'Imagewriter n'existent pas.

Il ne nous reste donc qu'une solution, et il a bien fallu que j'y passe : se prendre par la main, et éditer, à l'aide de nos éditeurs de fontes, les couples de fontes caractère par caractère. Ca prend énormément de temps. Mais ça marche.

❄ *Fabriquer ses fontes*

Soyons francs : aucune des fontes héritées du Mac, qu'il s'agisse des fontes mises par Apple sur les disquettes système du GS, des fontes mises par Beagle sur les disquettes SuperFonts, ou des fontes domaine public diffusées par tout un chacun, ne peut être utilisée **telle quelle** pour imprimer sur l'Apple // (8 bits ou GS). La conversion d'une fonte Mac nous donne une base de travail : à nous de l'éditer pour en faire vraiment une fonte Apple //.

Voici donc quelques conseils pour l'éditeur de fontes que vous serez inévitablement un jour : ils m'ont été dictés par l'expérience.

■ Partez d'une fonte Mac qui soit conçue pour l'imprimante, et non pour l'écran du Mac : les fontes correspondant à celles de la LaserWriter (*Helvetica*, *Times*, *Courier*) ou une fonte comme *New York* font de bons points de départ. Ne faîtes strictement aucune confiance aux nombres affectés à ces fontes, et partez de leur hauteur réelle.

Renommez ces fontes, et changez leur numéro de famille : certains logiciels ont besoin, pour leur affichage écran, de la fonte Mac originale. Les changements que vous allez faire perturberaient leur affichage. Utilisez des numéros supérieurs à 32768, et vous serez tranquilles (le Macintosh n'a que huit bits pour numéroter ses fontes, d'où un beau bazar que le Font/DA Mover gère comme il peut).

■ Vérifiez tout sur papier, en imprimant des textes entiers avec votre fonte. Ne faîtes pas confiance aux suggestions de l'écran.

■ Faîtes attention aux blancs à laisser entre les caractères : le nombre de points n'est pas le même pour chaque caractère, et il doit être cohérent sur l'ensemble de la fonte. Attention aux caractères de ponctuation, qui sont toujours en fait suivis d'un espace dans les textes : laissez-leur du blanc à gauche, mais pas à droite. Veillez à ce que l'espace dur (Ascii 202) ait la même largeur que l'espace tout court. Ne faîtes pas un

espace (Ascii 32) trop large : si vous choisissez d'imprimer en justification totale, les alignements seront obtenus en élargissant les espaces.

■ Editez précautionneusement la fonte de taille double : c'est elle qui sera imprimée. La fonte de taille simple n'est qu'une vérification pour l'écran, seules les dimensions des caractères sont importantes, et non leur aspect. Vous aurez donc avant tout à vérifier les largeurs, caractère par caractère.

■ Veillez bien à ce que votre fonte de taille double ait une hauteur paire, et que chacun des caractères ait également une largeur paire (il n'y a pas de demi-point). Ne pensez pas vous en tirer en ajustant les blancs entre caractères, il n'y a généralement qu'une solution, qui est de redessiner le caractère !

Si vous faîtes des fontes en tiercé, type *12/24/48*, alors commencez par éditer la fonte 48 : sa hauteur, et la largeur de chacun des caractères, devront obligatoirement être des multiples de 4. Bon courage !

■ Dans la fonte double, remontez d'un point la ligne de base par rapport à la base réelle. C'est nécessaire pour avoir un soulignement correct sur papier : sans quoi QuickDraw imprimera le trait de soulignement trop près de la base du caractère.

■ Si vous faîtes des fontes d'icônes, en capturant des dessins Print Shop par exemple, et que vous les utilisez en même temps que les fontes Mac, alors il faut «macintoshiser» vos fontes icônes, en doublant verticalement (verticalement seulement) chaque caractère : vous imprimerez en mode «Condensé», ce qui rétablira la dimension originelle.

❄ *Le gras double*

Avec tout ça, ça marche. Il ne reste plus, à ma connaissance, qu'un pépin : le mode gras, et les autres modes (ombré, etc). Quand QuickDraw affiche sur l'écran en mode gras, il rajoute un point (pas un demi-point), en utilisant la fonte simple. Quand on imprime en mode gras, il rajoute toujours un point (pas deux), mais à la fonte double cette fois-ci : il y a donc un décalage inévitable, puisque l'impression se fait en divisant par deux.

Là aussi, il y a une solution, bien sûr, mais elle n'est pas évidente : ce sont les fontes «pré-stylées». Si vous voulez que la fonte double qui sera imprimée soit aussi, en mode gras, le double de la fonte simple, il faut créer, à l'éditeur de fontes, une fonte «gras-double» (oui, je sais, c'est un très mauvais jeu de mots, mais je n'ai pas pu résister) et une fonte «gras-simple», et les ajuster ensuite, caractère par caractère.

Le Font Manager permet cela, précisément pour les quelques cas où le travail mécanique de QuickDraw ne suffit pas. Il existe dans l'en-tête des fontes GS deux octets de «style» qui sont là pour lui signaler de quel style est cette fonte. Voir plus loin dans l'article, la documentation de Fontasm 2.0 et le ToolBox Reference pour toutes précisions.

Tiens, une petite colle à ce propos, si vous utilisez TimeOutSuperFonts pour AppleWorks-8 (lequel contient son petit QuickDraw 8 bits) : SuperFonts utilise-t-il les fontes pré-stylées ?

❄ *Soyez courageux : ça en vaut la peine...*

«Oh là là, quel travail», me direz-vous. Oui, c'est comme ça. Ecran et imprimante ne sont pas la même chose, il n'y aura jamais de «Wysiwyg» parfait. Et quand on veut remédier à la chose avec des écrans de résolution supérieure et des cartes graphiques, c'est plus cher et c'est encore plus compliqué.

Mais ce n'est pas tant de travail que vous le craignez, car vous n'avez pas besoin de centaines de fontes : quatre ou cinq couples de fontes suffisent largement. Pour la variété, jouez sur les styles dans la même fonte plutôt que de multiplier les fontes. Au fur et à mesure de vos impressions, vous ajusterez vos fontes favorites, que vous maîtriserez totalement et personnaliserez : c'est cela, l'Apple //...

De plus, l'ami Stephan Hadinger, éminent auteur de ToolBox Mag, nous a fourni avec son Fontasm 2.0 l'outil parfait pour ce travail. Utilisez en particulier ses capacités de rapport sur les fontes.

Surtout, n'oubliez pas que le jeu en vaut la chandelle : avec une Imagewriter, des rubans fraîchement encrés, AppleWorks-GS et de bonnes fontes, on fait des papiers qui ont une sacrée allure.

Ce système de la double réduction, instauré par le driver de GS/OS comme par TimeOut SuperFonts, permet des résultats sur papier supérieurs à ceux de Publish-It, qui utilise les fontes en taille simple, supérieurs aussi à ceux de Multiscribe/Beaglewrite 8 bits, et de niveau égal et souvent supérieur à ceux de Print Shop.

Il n'est plus guère possible, du point de vue de la qualité d'impression avec l'ImageWriter, de faire mieux que GS/OS (un seul progrès serait envisageable : une option pour forcer une impression unidirectionnelle, pour ceux qui ne comptent pas le temps et veulent à n'importe quel prix la qualité maximale).

A l'heure où nous mettons en page cet article, nos agents à Cupertino nous ont fait parvenir des drivers en bêta version, plus rapides et permettant l'impression en mode unidirectionnel.

Les progrès possibles dépendent désormais des applications (quel logiciel va nous faire de la micro-justification ?), et de la qualité des fontes.

Avec la versatilité des fontes, si l'on respecte l'ensemble des conditions énoncées ci-dessus, les impressions envoyées par AppleWorks-GS dans tous ses modules (tableur et base de données aussi bien que PAO et traitement de textes) sont d'une qualité qui n'est surpassée que par la Laser. Je n'ai vu aucun papier sortant d'une imprimante 9 aiguilles, quelle qu'elle soit et quels que soient l'ordinateur et le logiciel qui la pilotent, qui soit supérieur à ce que fait AppleWorks-GS sur ImageWriter. Le seul à l'égaler systématiquement, on s'en doutait un peu, c'est le Mac (à condition qu'il ait de bonnes fontes, lui aussi, et qu'on y choisisse aussi le meilleur mode d'impression)...

❄ ...Ou bien laissez ToolBox rouler pour vous !

Puisque ni Apple ni aucun éditeur n'a su faire les **vraies fontes pour l'ImageWriter** dont nous avons besoin, ToolBox a décidé de s'y coller. Car de quoi avons-nous besoin ?

Pas d'une variété infinie de fontes fantaisistes et belles sur l'écran, mais de jeux échelonnés des mêmes fontes : ce qu'il nous faut, ce sont quelques familles de fontes seulement, mais en plusieurs tailles, présentées sous formes de couples *fonte simple/fonte double*, et qui donnent un affichage correct dans tous les styles, particulièrement en gras et en italiques.

Vous trouverez donc sur la disquette de ce numéro (sous-catalogue `/JYB.FONTES`) quelques-unes des fontes dont je me sers couramment pour l'écran et l'ImageWriter.

Tout se suite une remarque : ne venez pas nous dire qu'il manque tel ou tel caractère, que vous auriez préféré un autre dessin pour tel caractère, etc. Faîtes comme moi, éditez une copie de la fonte en question pour la mettre à votre goût.

Ne demandez pas non plus de quelles fontes je me suis inspiré : j'ai pioché dans mes 20 disquettes de fontes GS, mélangé le tout, copié, collé, retravaillé, redimensionné, redessiné, etc. Ceci est un résultat assez stable, mais encore provisoire : je ne cesse pas d'ajuster mes fontes.

Ne dîtes pas non plus que ces fontes ne sont pas magnifiques sur l'écran : elles sont destinées uniquement à permettre une impression de qualité sur l'ImageWriter, avec les options 'Condensé' et 'Qualité Supérieure' (Better Text) du driver d'ImageWriter. L'écran du GS ne sert que de contrôle.

Les trois fontes *Shaston* sont réservées à l'affichage écran : tapez vos textes sous *Shaston*, la fonte la plus facilement discernable sur l'écran du GS, et choisissez ensuite seulement, avant d'imprimer, les fontes pour l'imprimante.

Shaston 16 est la même que celle des disques système Apple, mais complète jusqu'à l'Ascii 255. *Shaston 09* et *Shaston 10* sont en fait deux variantes de la Shaston 8 de la Rom. Toutes les deux sont cependant complètes (*Shaston 8* étant incomplète). *Shaston 09* est une fonte proportionnelle, mais qui admet le soulignement, à la différence de *Shaston 8*. *Shaston 10* est une fonte non-proportionnelle, où tous les caractères ont la même largeur : elle sert à afficher sur écran les tableaux, listings, etc. avant de les imprimer en *Courier*.

Les fontes *Meudon* sont des fontes pour l'ImageWriter. Leur dessin est acceptable, mais elles ont surtout l'avantage de permettre d'imprimer en cinq tailles dans la même fonte : *12*, *13*, *14*, *16*, *24*.

Les fontes *Bobigny* sont mes fontes les plus courantes. J'écris régulièrement en *Bobigny 12*, avec les notes en *Bobigny 09*.

Attention à bien mettre dans votre système toutes les variantes des fontes *Bobigny 12* et *24* : c'est là que j'ai utilisé le truc des fontes 'pré-stylées', pour ajuster les caractères gras (suffixe .G) et les italiques (suffixe .I), et corriger les soulignés (suffixe .S).

A vous de faire, si vous aimez ces styles, les styles Contour (Outline) et Ombré (Shadow). L'octet de style le permet, et QuickDraw gère tout cela impeccablement. Il permet aussi de faire des styles combinés : Gras + Ombré, Italiques + Contour + Souligné, etc.

Le seul inconvénient est que certains logiciels, ignorant l'octet de style, vous proposent plusieurs fois la même fonte *Bobigny*, vous laissant deviner son style. Il suffirait d'un petit truc pour régler cela : forcer l'affichage des noms des fontes dans le menu Fontes à se faire avec ces mêmes fontes. Tiens, à propos, voilà une belle Init à programmer : allez, qui s'y colle ?

En tout cas, je n'approuve pas la prudence du ToolBox Reference, qui conseille de laisser faire QuickDraw. Comme précisément, avec Fontasm, on part de la fonte telle que QuickDraw l'a transformée dans le style demandé, on ne peut guère que faire mieux, surtout en testant systématiquement sur papier.

Sans doute Apple craint-il de voir se multiplier les fontes BitMap dans le Dossier Système : s'il y a une fonte bitmap pour chaque style et chaque combinaison de styles de chaque taille de chaque fonte, une seule fonte occupera plusieurs mégas (et QuickDraw ne servira plus à rien).

Mais il n'est pas question de cela : c'est seulement pour les styles utilisés le plus souvent de nos fontes usuelles que cela a un sens.

Le couple 'Bobigny.Maj' (Bobigny Majuscules) 12/24 sert à imprimer dans le style dit 'Petites Capitales', où les minuscules sont représentées par des majuscules plus petites.

Ce style n'étant pas un style QuickDraw, il n'est pas représentable dans l'octet de style : il m'a donc fallu faire une nouvelle famille de fonte.

Sur le même principe, vous pouvez créer d'autres styles non-QuickDraw, par exemple une '*Bobigny* étroite'.

La disquette ToolBox tend ses sillons à vos créations de variantes de Bobigny ou autres : relisez les conseils du point en page 12, lancez Fontasm 2.0, et vous jalouserez moins la LaserWriter et son PostScript !

Dessin digitalisé avec Quickie et imprimé sur LaserWriter II à partir d'AppleWorks GS

Un 'StartUpTools' pour les accessoires

Patrick Desnoues

Le nouvel appel 'StartUpTools' du système 5.0 a réduit considérablement la longueur des listings. Hélas, on ne peut l'utiliser dans un accessoire, qui ignore par définition quels outils sont actifs au moment où il démarre, et doit laisser en sortant ces mêmes outils exactement dans l'état où il les a trouvés (pour les autres accessoires et pour l'application).

Voici donc une routine standard d'initialisation et de fermeture des outils pour utilisation dans un NDA (éventuellement CDA ou Init, bref tout ce qui n'est pas une application). La méthode classique étant :

```
Init ---------------
        OutilStatus
        pas en mémoire -> LoadOutil
        Non Actif      -> Startup Tool
     Autre outil
     etc .......
Fermeture ----------
        Si Load Tool     -> Unload Tool
        Si StartupOutil -> ShutDown Tool
     Autre outil
     etc .......
```

Plus on utilisait d'outils, plus le programme se rallongeait...

Cette routine est inspirée de la méthode utilisée depuis la version 5.0 du système par l'appel 'StartUpTools', à savoir le recours à une table de paramètres très simple à utiliser :

```
TableOutil   Anop
             dc i'ResourceManager , 0,    $11, $100'
             dc i'Scheduler       , 0,     0, $200'
*            dc i'SoundTool       , 0, $0102, $301'
*            dc i'Adb             , 0,     0, $201'
*            dc i'Sane            , 0, $0102, $202'
             dc i'IntegerMath     , 0,     0, $100'
*            dc i'TextTool        , 0,     0, $201'
*            dc i'SystemLoader    , 0,     0, $300'
*            dc i'ACE             , 0, $0102, $101'
*            dc i'MidiTool        , 0, $0303, $103'
*            dc i'ScrapManager    , 0,     0, $300'
             dc i'ListManager     , 0,     0, $301'
             dc i'FontManager     , 0, $0103, $301'
             dc i'QDAuxiliary     , 0,     0, $301'
             dc i'LineEdit        , 0, $0103, $301'
*            dc i'PrintManager    , 0, $0203, $300'
*            dc i'StandardFile    , 0, $0103, $301'
*            dc i'TextEdit        , 0, $0103, $101'
TableOutilEnd  Anop
```

Pour chaque outil, cette table nécessite 4 paramètres :

① Le nom de l'outil

② Une variable sur 2 octets permettant de stocker l'état de l'outil :

- `$0000` : Outil non actif -> Startup
- `$B123` : Outil non actif + LoadOneTool
- `$xxxx` : Outil actif

③ Une valeur permettant d'informer la commande Startup sur le nombre de paramètres à lui passer :

- `$0` : Aucun paramètre
- `$1` : UserId
  + `$0010` : Si cet outil est le resource manager
- `$2` : Page zéro
- `$3` : UserId + Page zéro
  + `$0100` : 1 page zéro nécessaire
  + `$0200` : 2 pages zéro nécessaires
  + `$0300` : 3 pages zéro nécessaires

④ La version minimum de l'outil à charger :
`v1.01 -> $0101`

Un plan pratique

Le fichier 'InitTools' comprend toute les routines nécessaires à inclure dans vos programmes. La table des outils étant déjà créée, il suffit de supprimer le caractère '*', pour autoriser l'outil et de l'ajouter pour l'invalider.

Dans votre programme deux appels :

`Jsr InitOutils`	(pour l'initialisation)
`Jsr ShutDownToolAp`	(pour la fermeture)

La fermeture se fait évidemment dans l'ordre inverse des outils.

Autres paramètres :

- 'UserID' : variable ramenée par MemoryStartup ou mieux GetUserID.
- Le sous-programme utilise 4 variables en page zéro :

`'Handle'`	(Long : 4 octets)
`'Var1'`	(Word : 2 octets)
`'Var2'`	(Word : 2 octets)
`'Var3'`	(Word : 2 octets)

Certains outils (inutiles à initialiser dans un NDA) ne peuvent employer cette méthode :

- MemoryManager
- Quickdraw
- EventManager

Egalement :

- NoteSynthetizer
- NoteSequencer

Ceci est dû à des paramètres différents employés par la commande startup.

Un cas particulier : En cas de LoadOneTool (TextEdit), la commande UnloadOneTool refuse de le purger (?).

Note importante :

① Vous activez un NDA utilisant TextEdit (par exemple 'WriteIt'). Ce NDA va initialiser TextEdit.

② Vous activez un autre NDA utilisant TextEdit. Pour lui, 'TextEdit' étant actif, on ne peut de nouveau l'initialiser.

③ Vous fermez le premier NDA, qui va fermer TextEdit, puisqu'il l'a initialisé.

④ La prochaine fois que le deuxième NDA va employer un outil du TextEdit, le plantage guette car l'outil n'est plus actif !

Pour l'utilisateur : il faut donc fermer les NDA dans l'ordre inverse de leur ouverture.
Pour le programmeur : Tester le 'Status' des outils importants avant d'utiliser une de leurs fonctions.

Un NDA nommé 'ToolBox' (ça tombe bien) à titre de démo utilisant cette routine se trouve sur la disquette. Il affiche la version de tous les outils se trouvant sur le volume de Boot.

Ce n'est évidemment pas le premier permettant cela, mais en plus il affiche un '*' après le nom de l'outil si celui-ci était actif au moment du chargement du NDA. Il présente aussi l'intérêt de montrer comment programmer un NDA avec des ressources (tout en permettant de renommer le NDA : merci à Yvan Kœnig). Tous les sources nécessaires sont sur la disquette.

NDLR : La vocation de la routine qui suit est d'être incorporée telle quelle dans vos programmes. Développeurs et programmeurs, sachez que cette incorporation est libre et gratuite. Un petit mot gentil quelque part dans votre programme pour P. Desnoues et pour ToolBox Mag nous ferait bien plaisir. Merci.

```
*************************************************************
*
*   Initialisation et ShutDown des outils pour utilisation
*   avec un NDA
*   Assembleur APW/Orca
*   Patrick Desnoues - Toolbox Mag - Septembre 90
*
*************************************************************

InitOutils     START
               Using MyDatas

;-------------------------------------------
; Variables Page zéro                       |
; ......................                    |
; Handle  (Long) utilisé par NewHandle      |
; Var1    (Word) <- Pointeur TableOutil     |
; Var2    (Word) <- Numéro Outil            |
;-------------------------------------------

ToolLocator     Equ $01
MemoryManager   Equ $02
Miscellaneous   Equ $03
Quickdraw       Equ $04
DeskManager     Equ $05
EventManager    Equ $06
Scheduler       Equ $07
SoundTool       Equ $08
ADB             Equ $09
Sane            Equ $0A
IntegerMath     Equ $0B
TextTool        Equ $0C
WindowManager   Equ $0E
MenuManager     Equ $0F
ControlManager  Equ $10
SystemLoader    Equ $11
QDAuxiliary     Equ $12
PrintManager    Equ $13
LineEdit        Equ $14
DialogManager   Equ $15
ScrapManager    Equ $16
StandardFile    Equ $17
NoteSynthetizer Equ $19
NoteSequencer   Equ $1A
FontManager     Equ $1B
ListManager     Equ $1C
ACE             Equ $1D
ResourceManager Equ $1E
MidiTool        Equ $20
TextEdit        Equ $22

ToolBox         Equ $E10000

Startup         Equ $0200
ShutDown        Equ $0300

Status          Equ $0600

*----------------------------------------

ParUserID       Equ 1
ParDPage        Equ 2
ParUserDPage    Equ 3

*----------------------------------------
* Status = $B123 -> Outil : Load + Start
*        = 0     -> Outil : Start
*        > 0     -> Rien
;----------------------------------------

StatDrapeau     Equ $B123

***************  STARTUP  ***************

                Pushword £0
                _ResourceStatus
                Pla
                Sec
                Beq ErreurTool
```

```
                Pei <MyUserID
                _ResourceStartup

;------------------------------

                Ldy £0

LoopInitTools   Anop
                Lda TableOutil,Y

                Phy>>
                Jsr InitTool
                Ply<<
                Bcs ErreurTool

                Tya
                Adc £2*4  (4 param de 2 octets dans la table)
                Tay
                Cmp £NombreTool
                Bcc LoopInitTools

NoErreurInit    Anop
                Clc

ErreurTool      Anop

; Si c=1 A contient le code d'erreur à traiter par
  l'application (_ErrorWindow)

                Rts

****************** TOUS LES SHUTDOWN **********************

ShutDownToolAp  ENTRY
                Lda £NombreTool

LoopShutDownT   Anop
                Sec
                Sbc £2*4  (4 param de 2 octets dans la table)
                Tay
                Bmi EndSDTool
                Lda TableOutil,Y

                Phy>>
                Jsr ShutDownTool
                Pla<<
                Bra LoopShutDownT

EndSDTool       Anop
                _ResourceShutDown
                Rts

************ Sous programme ouverture des outils ************

InitTool        Anop

; A= TableOutil,Y

                Sta <Var2                Numéro Tool

                Iny
                Iny
                Sty <Var1

                Pushword £0
                Clc
                Adc £Status
                Tax
                Jsl ToolBox

;-------------------------------------------------------------
; c = 1 outil non chargé -> Load puis Startup
; On mettra la valeur du status à £$B123 (pourquoi pas celle
; la) après avoir chargé l'outil par LoadOneTool pour qu'à la
; fermeture du NDA on puisse utiliser la commande
; UnloadOneTool
;
; c = 0 : valeur retournée  = 0 -> outil non actif -> Startup
;                           <> 0 -> Ne rien faire
;-------------------------------------------------------------

                Bcs LoadTool

                Lda 1,s
                Bne EndInitTool
                Bra ToolStartup

;------------------------------

LoadTool        Anop
                Pei <Var2
                Ldy <Var1                Pagezero ?
                Iny
                Iny
                Iny
                Iny
                Lda TableOutil,Y         Version minimum
                Pha
                _LoadOneTool
                Bcs EndInitTool

                Lda £StatDrapeau
                Sta 1,s

;------------------------------

ToolStartup     Anop
                Ldy <Var1            Pagezero ?
                Iny
                Iny
                Lda TableOutil,Y     A=1 -> Pagezero
                And £$F0FF

; 0 = 0 Paramètre pour Startup
; 1 = UserID avant Startup
; 2 = Fixer une page zéro
; 3 = 2 + 3 : UserID + DPage

                Beq NoParametre      0 : Pas de paramètre

                Cmp £ParUserID
                Bne DPage
;------- Pour xxxStartup
                Pei <MyUserId        = 1 : Push UserID
                Bra NoParametre

DPage           Anop
                Cmp £ParDPage
                Beq FixPageZero

                Cmp £ParUserDPage

;------ Pour xxxStartup
                Pei <MyUserId        = 3 : Push UserID

FixPageZero     Anop
                Ldx £0

                LongResult

                Phx                  Taille à allouer Haute
                Lda TableOutil,Y
                And £$0F00           Taille à allouer Basse
                Pha

                Pei <MyUserId
                Pushword £%1100000000000101   Attributes
                Phx                           Pointeur banc 0
                Phx
                _NewHandle
                Plx
                Ply
                Bcs EndInitTool

                Stx <Handle
                Sty <Handle+2

;------ Pour xxxStartup
                Lda [Handle]         2 et 3 : Push DPage
                Pha

NoParametre     Anop
                Lda <Var2
                Clc
                Adc £Startup
                Tax
                Jsl ToolBox

EndInitTool     Anop
                Tax              Sauvegarde le code d'erreur

                Ldy <Var1
                Pla              Résultat Status
                Sta TableOutil,Y

                Txa              Code d'erreur si c=1
                Rts

************ Sous-programme Fermeture des outils ************

ShutDownTool    Anop

; A= TableOutil,Y

                Sta <Var1

                Iny
                Iny
```

```
                Clc
                Adc £ShutDown
                Tax

                Lda TableOutil,Y
                Bne LMSD2

                Jsl ToolBox
                Rts

LMSD2           Anop
                Cmp £StatDrapeau
                Beq LMSD1
                Rts

LMSD1           Anop
                Jsl ToolBox
                Pei <Var1
                _UnloadOneTool
                Rts

*************************************************************
;
;               dc i'NomOutil, Status, Paramètre, VersionMini'
;                           = ds 2
;                           =0 : Aucun paramètre pour Startup
;                           =1 : UserID
;                           +10 = Resource Manager
;                                   =2 : DPage
;                           =3 : UserID + DPage
;                                       +100 = 1 Page zéro
;                                       +200 = 2 Page zéro
;                                       +300 = 3 Page zéro
;
*************************************************************

NombreTool      Equ TableOutilEnd-TableOutil

TableOutil      Anop
                dc i'Scheduler       , 0,      0, $200'
*               dc i'SoundTool       , 0, $0102, $301'
*               dc i'Adb             , 0,      0, $201'
*               dc i'Sane            , 0, $0102, $202'
                dc i'IntegerMath     , 0,      0, $100'
*               dc i'TextTool        , 0,      0, $201'
*               dc i'SystemLoader    , 0,      0, $300'
*               dc i'ACE             , 0, $0102, $101'
*               dc i'MidiTool        , 0, $0303, $103'
*               dc i'ScrapManager    , 0,      0, $300'
                dc i'ListManager     , 0,      0, $301'
                dc i'FontManager     , 0, $0103, $301'
                dc i'QDAuxiliary     , 0,      0, $301'
                dc i'LineEdit        , 0, $0103, $301'
*               dc i'PrintManager    , 0, $0203, $300'
*               dc i'DialogManager   , 0,      1, $302'
*               dc i'StandardFile    , 0, $0103, $301'
*               dc i'TextEdit        , 0, $0103, $101'
TableOutilEnd   Anop
*************************************************************
;
; Paramètre avant commande Startup
;
; $01 : ToolLocator     : 0
; $02 : MemoryManager   : WordResult - Tool - Pla
; $03 : Miscellaneous   : 0
; $04 : Quickdraw       : Dpage *3 / MasterSCB / MaxWith /
                          UserID
; $05 : DeskManager     : 0
; $06 : EventManager    : Dpage *1 /QueueSize/XMin/XMax/
                          YMin/YMax/ UserID
; $07 : Scheduler       : 0
; $08 : SoundTool       : DPage *1
; $09 : ADB             : 0
; $0A : Sane            : Dpage *1
; $0B : IntegerMath     : 0
; $0C : TextTool        : 0
; $0E : WindowManager   : UserID
; $0F : MenuManager     : UserID / DPage *1
; $10 : ControlManager  : UserID / DPage *1
; $11 : System Loader   : 0
; $12 : QDAuxiliary     : 0
; $13 : PrintManager    : UserID / DPage *2
; $14 : LineEdit        : UserID / DPage *1
; $15 : DialogManager   : UserID
; $16 : ScrapManager    : 0
; $17 : StandardFile    : UserID / DPage *1
; $19 : NoteSynthetizer : Rate / MyRoutine
; $1A : NoteSequencer   : DPage / UserID / Rate / Increment
; $1B : FontManager     : UserID / DPage *1
; $1C : ListManager     : 0
; $1D : ACE             : DPage *1
; $1E : ResourceManager : UserID - cas particulier : Pas de
                          LoadOneTool
; $20 : MidiTool        : UserID / DPage *3
; $22 : TextEdit        : UserID / DPage *1
;
*************************************************************
;
; 0 Paramètre
; -----------
; $01 : ToolLocator     : 0
; $03 : Miscellaneous   : 0
; $05 : DeskManager     : 0
; $07 : Scheduler       : 0
; $09 : ADB             : 0
; $0B : IntegerMath     : 0
; $0C : TextTool        : 0
; $11 : System Loader   : 0
; $12 : QDAuxiliary     : 0
; $16 : ScrapManager    : 0
; $1C : ListManager     : 0
;
; UserID
; ------
; $0E : WindowManager   : UserID
; $15 : DialogManager   : UserID
;
; DPage
; -----
; $08 : SoundTool       : DPage *1
; $0A : Sane            : Dpage *1
; $1D : ACE             : DPage *1
;
; UserID + DPage
; --------------
; $0F : MenuManager     : UserID / DPage *1
; $10 : ControlManager  : UserID / DPage *1
; $13 : PrintManager    : UserID / DPage *2
; $14 : LineEdit        : UserID / DPage *1
; $17 : StandardFile    : UserID / DPage *1
; $1B : FontManager     : UserID / DPage *1
; $20 : MidiTool        : UserID / DPage *3
; $22 : TextEdit        : UserID / DPage *1
;
; Cas Particulier
; ---------------
; $1E : ResourceManager : UserID - mais pas de LoadOneTool
;
; Cas Particulier à initialiser de façon classique
; ------------------------------------------------
; $02 : MemoryManager   : WordResult - Tool - Pla
; $04 : Quickdraw       : Dpage *3 / MasterSCB / MaxWith /
                          UserID
; $06 : EventManager    : Dpage *1 /QueueSize/XMin/XMax/
                          YMin/YMax/ UserID
; $19 : NoteSynthetizer : Rate / MyRoutine
; $1A : NoteSequencer   : DPage / UserID / Rate / Increment

                        END
```

Attention !

ToolBox Mag est composé inséparablement d'une revue sur papier et d'une disquette pour ordinateur Apple IIGS.

Certains avertissements légaux, modes d'emploi, etc. peuvent être placés sur la disquette : ils n'en sont pas moins écrits dans ToolBox Mag, dont ils font partie intégrante.

Music Juke-box

Stephan Hadinger

Le programme de musique le plus puissant sur GS au moment où j'écris ces lignes est sans aucun doute Music Studio™, avec ses 15 instruments simultanés et un son polyphonique sur 15 voix.

Mais vous êtes nombreux à regretter qu'il soit impossible de récupérer vos belles musiques pour les inclure dans vos propres programmes. Voilà qui est chose faite.

Music Juke-box : oui et comment !

Music Juke-box est un programme qui vous permet de sonoriser vos programmes en GS/OS avec n'importe quelle musique créée sur Music Studio™. Il vous permet de charger jusqu'à 40 morceaux en mémoire et prélève la mémoire via le Memory Manager, il n'est donc limité qu'à la taille de votre carte d'extension de mémoire.

Vous pourrez ensuite jouer ces morceaux un par un, les uns à la suite des autres ou dans l'ordre que vous aurez choisi, tout en conservant un contrôle en temps réel pour une synchronisation avec votre programme.

Comme vous l'avez compris, les morceaux sont exécutés en même temps que votre programme, par le système d'interruptions, moyennant un léger ralentissement. La puissance et la facilité d'utilisation de Music Juke-box vient du fait qu'il est constitué d'un nouvel outil, le premier outil GS complet à être publié : le TOOL069.

Music Juke-box –Mjuke en abrégé– est compatible avec toutes les options de Music Studio. Il respecte tous les types de notes –triplets, liaisons, accents– les répétitions de parties de morceau, le volume sonore global, le tempo, toutes les clefs et tous les instruments, même en stéréo. Les instruments utilisés sont exactement les mêmes que ceux définis dans vos morceaux par Music Studio. Ainsi tout morceau composé avec Music Studio sera joué de manière absolument identique par Music Juke-box.

Music Juke-box est constitué de deux programmes : un programme de conversion et l'outil à proprement parler.

Conversion : MUSICONV

Tous les morceaux doivent être en premier lieu convertis du format Music Studio au format Juke-box via le programme ***MUSICONV***. Lancez-le par le Finder, puis choisissez l'option convertir dans le menu Fichier. Sélectionnez la musique à convertir et sauvez la version au format Juke-box avec le suffixe "`.MJK`". Attention lors de cette conversion, le fichier *Wavebank* utilisé par le morceau doit être dans le même directory que le morceau lui-même. Le fichier *Wavebank* est le gros fichier de 130 blocs contenant les différentes formes d'onde des instruments tels que *ROCK.WBNK*, *JAZZCOMBO.WBNK*, etc.

Cette opération de conversion permet de traduire les musiques dans un format beaucoup plus facilement exploitable, mais qui risque d'être légèrement plus encombrant que le morceau original. Le format *Juke-box* est décrit plus en détail à la fin de l'article.

L'option *Ecouter* permet d'entendre le morceau converti, et l'option Stop permet d'arrêter cette écoute. Le menu *Entraînement* vous offre la possibilité d'essayer la plupart des commandes de MJuke.

Si les outils *Note Synthesizer* (TOOL025) ou *MJuke* (TOOL069) sont absents du *sous-directory SYSTEM/TOOLS/* de la disquette de démarrage, le programme de conversion fonctionnera, mais les options *Ecouter*, *Stop* et le menu *Entraînement* seront désactivées.

L'outil : MJuke

Votre programme peut maintenant jouer toutes les musiques converties par l'intermédiaire de ***MJuke*** (l'outil 69). Pour qu'il soit utilisable par votre programme, placez-le dans le *sous-directory SYSTEM/TOOLS/* de votre disquette de démarrage.

Les listings 1, 2 et 3 sont les interfaces pour l'outil 69, utilisables respectivement en Assembleur, C et TML Pascal. Si vous utilisez l'Assembleur, ajoutez `MJUKE.MACROS` aux macros de votre programme en tapant "`MACGEN prog.SRC prog.MACROS 2/AINLUDE/M16.= M16.MJUKE`".

Si vous utilisez le C, placez "`#include <mjuke.h>`" au début de votre programme. Si vous utilisez le TML Pascal, compilez `MJUKE.PAS` en `MJUKE.USYM` puis ajoutez "MJUKE" à la liste qui suit "`USES`" dans votre programme.

MJuke vous offre maintenant 14 nouvelles commandes :

```
MJStartUp : ($0245)
        entrée  : MemID     : integer
                  ErrHndl   : integer
        sortie  : -
```

MJStartup initialise le Juke-box. Cet appel doit être fait avant tout autre appel à Mjuke et de préférence vers le début du programme, juste après toutes les autres initialisations d'outils. Il n'est pas nécessaire d'initialiser ou de charger le *Sound Manager* et le *Note Synthesizer*, *MJStartup* le fait automatiquement. En revanche si le *Note Synthesizer* (TOOL025) est absent, MJuke renvoie une erreur fatale.

Les paramètres à envoyer sont : l'identificateur renvoyé par *Memory Manager* au début du programme ('*MemID*'), et une valeur booléenne (*TRUE* ou *FALSE*) dans '*ErrHndl*' ; *FALSE* rendra l'outil 69 standard, c'est-à-dire que les erreurs seront renvoyées telles quelles à l'application, option obligatoire avec des programmes en mode texte ; en revanche *TRUE* ordonnera à MJuke de traiter lui-même les erreurs en affichant une fenêtre d'alerte, ce qui est bien pratique et qui décharge l'application de cette tâche ; dans ce cas-là, les outils *Window Manager*, *Dialog Manager*... doivent être initialisés, sous peine de plantage.

Exemple :

```
[Assembleur]
        PushWord MyMemoryID
        PushWord #$FFFF     ; (TRUE)
        _MJStartUp
[langage C]
        MJStartup(MyMemoryID, TRUE);
----------------
erreur possible:  $4501 = MJuke a déjà été initialisé.
----------------
```

```
-MJShutDown : ($0345)
        entrée : -
        sortie : -
```

Le morceau courant est arrêté, toute la mémoire réservée par MJuke est libérée, les outils *Sound Manager* et *Note Synthesizer* sont fermés et MJuke est désactivé. Cet appel se fait normalement à la fin du programme, juste avant la fermeture de tous les autres outils.

Exemple :

```
[Assembleur]
        _MJShutDown
[langage C]
        MJShutDown();
```

```
-MJVersion : ($0445)
        entrée : place pour résultat        : integer
        sortie : Version  : integer
MJVersion renvoie la version de MJuke (normalement $0101).
```

Exemple :

```
[Assembleur]
        PushWord #$0000
        _MJVersion
        pla       ; Accumulateur = numéro de version
[langage C]
        vers = MJVersion();
```

```
-MJStatus : ($0645)
        entrée : place pour résultat        : integer
        sortie : Status   : integer
```

MJStatus renvoie *TRUE* (*$FFFF*) si MJuke a déjà été initialisé, *FALSE* (*$0000*) dans le cas contraire.

Exemple :

```
[Assembleur]
        PushWord #$0000
        _MJStatus
        pla       ; Accumulateur = état de MJuke
[langage C]
        etat = MJStatus();
```

```
-MJMount : ($0945)
        entrée : Name     : pointeur
                 Prefix   : integer
        sortie : -
```

MJMount charge en mémoire la musique dont le nom –chaîne de type PASCAL donc précédée par la longueur de la chaîne– est passé par son adresse '*Name*'. Le nom doit obligatoirement se terminer par "*.MJK*", si vous mettez autre chose, les 4 dernières lettres du nom sont remplacées par "*.MJK*".

On peut charger jusqu'à 40 morceaux en mémoire. A chaque morceau est attribué un numéro correspondant à son rang de chargement –1 pour le premier, 2 pour le second, etc– c'est ce numéro qui servira à choisir les morceaux qui devront être joués.

Le fichier Wavebank est automatiquement chargé avec le premier morceau. Dès lors tous les autres morceaux devront utiliser ce même fichier Wavebank ; par exemple si le premier morceau utilise des instruments *ROCK*, les autres morceaux chargés ne pourront pas utiliser des instruments *JAZZ*. On passe dans 'Prefix' le numéro de préfixe (entre 0 et 7) où MJuke doit aller chercher le fichier Wavebank. Ce numéro de préfixe correspond au préfixes 0/... à 7/... de GS/OS. Le préfixe 0 correspond au directory par défaut, c'est celui qui est utilisé par l'outil Standard File. Le préfixe 1 correspond au directory contenant votre programme. Ce paramètre est important car lorsqu'un programme est exécuté directement par GS/OS sans passer par le Launcher ou le Finder, le préfixe 0 n'est pas initialisé et son utilisation renverrait l'erreur "Invalid Pathname Syntax". Dans ce cas-là il faut soit fixer soi-même le préfixe 0, soit plus simplement placer le fichier Wavebank dans le même directory que votre programme et le charger par le préfixe 1, c'est-à-dire mettre le

paramètre 'Prefix' à 1. Pour la même raison, il est souvent préférable de faire précéder le nom des morceaux par "1/". Les préfixes 3 à 7 peuvent être utilisés librement par votre programme, le 2 est réservé par GS/OS (cf. docs techniques d'Apple).

Exemple :

```
[Assembleur]
song     str      "1/TESTSONG.MJK"
         .....
         PushLong #song
         PushWord #1
         _MJMount
[langage C]
         MJMount("\p1/TESTSONG.MJK", 1);
         /* chaîne de type Pascal, indiquée par \p */
----------
```

Erreurs possibles :

$4502 : Fichier Juke-box incorrect, MJuke ne reconnait pas là un fichier au format Juke-box.

$4503 : Fichier Wavebank non chargé, le fichier Wavebank actuellement chargé ne correspond pas à celui du morceau à charger.

$4504 : Fichier Wavebank introuvable.

$4505 : Fichier Wavebank incorrect, MJuke ne reconnait pas le fichier Wavebank.

$4506 : Pas plus de 40 musiques.

$4507 : Numéro de préfixe incorrect, ce numéro doit être compris entre 0 et 7.

```
-MJPlay : ($0A45)
        entrée :       Morceau: integer
        sortie :       -
```

MJPlay joue le morceau dont le numéro est passé par 'Morceau'. Si aucun morceau n'est chargé MJPlay ne fait rien. Si le morceau n'existe pas ou n'est pas chargé, MJPlay joue le premier morceau. Si l'on fait 'Morceau' = $FFFF ou 'Morceau' = -1 (ce qui revient au même), MJPlay rejoue le morceau courant. Si l'on fait Morceau = 0, MJPlay joue le morceau suivant et revient au premier s'il est arrivé au dernier morceau. Dès que le morceau est fini, MJuke s'arrête et se met en attente.

Exemple :

```
[Assembleur]
        PushWord #$FFFF
        _MJPlay  ; rejoue le morceau courant
[langage C]
        MJPlay(0);        /* joue le morceau suivant */


-MJJuke : ($0B45)
        entrée : Morceau  : integer
        sortie : -
```

MJJuke s'utilise comme MJPlay, mais il enchaîne les morceaux les uns à la suite des autres ou dans un ordre prédéterminé par MJNext (voir plus loin), au lieu de n'en jouer qu'un seul.

Exemple :

```
[Assembleur]
        PushWord #0001
        _MJJuke
[langage C]
        MJJuke(1);        /* joue tous les morceaux chargés
en mémoire les uns à la suite des autres */


-MJCont : ($0C45)
        entrée : -
        sortie : -
```

MJCont continue l'exécution du morceau interrompu par MJStop.

Exemple :

```
[Assembleur]
        _MJCont
[langage C]
        MJCont();


-MJStop : ($0D45)
        entrée : -
        sortie : -
```

MJStop interrompt l'éxécution du morceau courant, jusqu'à MJCont, ce qui est utile avant un accès disque par exemple, pour ne pas entendre la musique ralentir et se saccader.

Exemple :

```
[Assembleur]
        _MJStop
[langage C]
        MJStop();


-MJWave : ($0E45)
        entrée : -
        sortie : -
```

MJWave permet de changer de Wavebank. Cet appel efface tous les morceaux en mémoire et permet de charger d'autres morceaux utilisant un autre fichier Wavebank. Cet ordre est donc normalement suivi de MJMount, et on recommence le chargement à partir du numéro 1. Si le nouveau fichier Wavebank est le même que celui des morceaux précédents, celui-ci n'est pas rechargé.

Exemple :

```
[Assembleur]
        _MJWave
[langage C]
        MJWave();


-MJNext : ($0F45)
        entrée : Adr      : pointeur
        sortie : -
```

MJNext permet d'installer une routine de gestion de l'enchaînement des morceaux en passant son adresse dans 'Adr', dans le cas d'un appel à MJJuke. Cette routine sera

appelée à la fin de chaque morceau en mode full native (*Acc* et *X/Y* en 16 bits) par un JSL, elle devra donc se terminer par un RTL et ne pas modifier l'environnement d'appel. Cette routine sera aussi appelée pour déterminer le morceau à jouer par *MJJuke(0)* ou *MJPlay(0)*. Lors de l'appel de la routine, le banc de données est quelconque donc si la routine est écrite en assembleur elle devra soit utiliser l'adressage long, soit fixer correctement le banc de données, si elle est écrite en C elle ne devra pas utiliser de variables globales, et si elle est écrite en TML Pascal elle devra être précédées de "*{$DefProc}*" et de "*{$LongGlobals+}*", puis suivie de "*{$LongGlobals-}*". Le numéro du morceau qui vient de se terminer est passé à la routine par la pile. La routine devra renvoyer par la pile le numéro du morceau qui doit suivre, une valeur de -1 ou 0 permet là aussi de rejouer le morceau ou de jouer le morceau suivant. Le listing 4 montre un exemple d'une telle routine en TML Pascal 1.00D, la version C se trouve dans le source *MUSIC.C*. Cette routine joue les morceaux dans l'ordre 1-3-2-4-...

Pour revenir à la gestion automatique par MJuke, c'est-à-dire jouer les morceaux les uns à la suite des autres, il suffit de faire 'Adr' = 0.

Exemple :

```
{Assembleur}
   PushLong   #routine
   _MJNext    ; impose "routine" à MJuke
   PushWord   #1
   _MJJuke    ; joue les morceaux à partir du premier
{langage C}
   MJNext(routine);   /* impose "routine" à MJuke */
   MJJuke(1); /* joue les morceaux à partir du premier */


-MJSong : ($1045)
  entrée :       place pour résultat        : integer
  sortie :       Song      : integer
```

MJSong renvoie dans '*Song*' le numéro du morceau courant ou du dernier morceau joué (entre 1 et 40).

Exemple :

```
{Assembleur}
        PushWord #$0000
        _MJSong
        pla       ; Acc = morceau courant
{langage C}
        song = MJSong();


-MJTimer : ($1145)
        entrée : place pour résultat        : integer
        sortie : Timer     : integer (non signé)
```

MJTimer renvoie dans *Timer* le temps en dixièmes de seconde qui s'est écoulé depuis le début du morceau courant. Cela permet par exemple de synchroniser l'image avec le son. Si le morceau courant est arrivé à sa fin et qu'aucun autre morceau n'a pris la relève (si on a utilisé *MJPlay* au lieu de *MJJuke*), la valeur renvoyée est -1 (ou *$FFFF*), valeur normalement jamais atteinte sauf si le morceau dure plus de 1h50 !

Attention si vous travaillez avec des entiers signés, le nombre qui suit 32767 est -32768, mais cela se produit seulement si le morceau dépasse 54 minutes !

Exemple :

```
{Assembleur}
        PushWord #$0000
        _MJTimer
        pla       ; Acc = temps écoulé
{langage C}
        temps = MJTimer();
```

Détails techniques

MJuke utilise le Note Synthesizer version 1.0 ou ultérieure et l'initialise à une fréquence d'interruptions de 100 Hz. La taille des musiques converties est limitée à 64 Ko, ce qui est largement suffisant. Si vous trouvez qu'une musique est jouée à un niveau sonore trop faible, modifiez directement le volume sonore global du morceau avec Music Studio. Il est conseillé d'ajouter environ une seconde de silence à la fin de chaque morceau afin d'éviter que l'enchaînement des morceaux soit trop brusque.

Les morceaux de Music Studio sont convertis dans le format MJuke dont l'organisation suit :

$0000-0003 : "*MJK2*" cette chaine de caractère est la signature des fichiers Juke-box, le chiffre 2 correspond au numéro de version.

$0004-0007 : ces 4 octets sont réservés pour un usage ultérieur.

$0008-0017 : ces 16 octets contiennent le nom du fichier Wavebank utilisé par le morceau (chaîne de type Pascal).

$0018-0035 : ces 15*2 octets représentent les adresses relatives (offsets) des définitions des 15 instruments utilisés par le morceau.

$0036 : cet octet est le volume sonore global du morceau.

$0037-xxxx : ce sont les notes du morceau.

$xxxx-yyyy : ce sont les caractéristiques des 15 instruments, dans le format utilisé par le Note Synthesizer.

Les notes sont normalement codées sur 4 octets. Le premier octet correspond au nombre de ticks qui doit s'écouler avant de jouer la note suivante, dans un accord cette valeur est nulle. Le deuxième octet contient le demi-ton ($00-7F) de la note, augmenté de $80 si la note est accentuée. Le troisième octet représente l'instrument par lequel la note doit être jouée, il est nul dans le cas d'un silence, et s'il est augmenté de $80 il représente alors une fin de note infinie (voir ci-dessous). Le quatrième octet contient la durée de la note en ticks.

Cependant si l'intervalle de temps avant la prochaine note dépasse `$FD` ou si la durée de la note dépasse `$FF`, les notes sont alors codées sur sept octets. Le premier octet est `$FF`, puis la note est décrite comme précédemment, les durées avant la prochaine note et de la note elle-même n'étant plus codées sur un mais sur deux octets.

Ce format long comporte deux nouveautés, si les trois premiers octets sont `$FFFFFF` alors le morceau s'achève, si la durée de la note est égale à `$FFFF` alors il s'agit d'une note infinie qui ne s'arrêtera que par une fin de note (`instrument > $80`), ces notes infinies correspondent aux liaisons entre notes dans Music Studio.

Enfin si le premier octet vaut `$FE`, cela signale le début d'une répétition, ces répétitions se situent toujours à la fin de la partie à répéter. Le second octet contient le niveau d'imbrication de la répétition dans d'autres répétitions (entre 0 et 15) ; il ne peut pas y avoir plus de 16 répétitons imbriquées dans un morceau. Le troisième octet contient le nombre de répétitions à effectuer auquel on soustrait 1 (entre 0 et 255). Les quatrième et cinquième octets contiennent le nombre d'octets à rebrousser pour atteindre le début de la partie à répéter.

Compilation avec APW 1.0

Pour compiler le TOOL069 créez d'abord le fichier de macros une fois pour toutes par "`MACGEN MJUKE.SRC MJUKE.MACROS 2/AINCLUDE/M16.= MJUKE.MAC`".

Ensuite tapez :

```
ASML MJUKE.SRC KEEP=MJUKE
RENAME MJUKE TOOL069
COMPACT TOOL069 -S
FILETYPE TOOL069 TOL
```

Pour compiler MUSICONV créez là aussi le fichier de macros une fois pour toutes par "`MACGEN MUSIC2.SRC MUSIC2.MACROS 2/AINCLUDE/M16.= MJUKE.MAC`".

A lire

Comme la routine InitTools de Patrick Desnoues, le TOOL069 est destiné à être utilisé tel quel dans vos programmes. Bien entendu, ToolBox Mag et Stephan Hadinger restent des propriétaires inflexibles des droits et du Copyright sur le TOOL69. Comment faire alors ?

Eh bien, il faut verser une redevance. Cette redevance consiste en... un petit mot gentil pour ToolBox Mag et Stephan Hadinger affiché à l'écran par votre programme. OK ?

Ensuite tapez :

```
COMPILE DEBUT.C KEEP=DEBUT
COMPILE MUSIC.C KEEP=MUSIC
ASSEMBLE MUSIC2.SRC
LINK 2/START MUSIC MUSIC2 DEBUT
KEEP=MUSICONV
COMPACT MUSICONV -S
FILETYPE MUSICONV S16
```

Faites attention à une erreur de la part d'Apple concernant les macros, les ordres `_AllNotesOff` du Note Synthesizer et du Note Sequencer portent le même nom. Pour éviter toute confusion MJUKE remplace cet ordre par _CouperNotes, défini dans MJUKE.MAC. Ecrivez de belles musiques et améliorez vos programmes, vous verrez que ce n'est pas difficile. Le listing 4 montre que quelque lignes de Pascal suffisent à exploiter la puissance du circuit Ensoniq de votre GS préféré. ■

Sources MJUKE

```
*
*       "MJUKE.SRC"
*
*       Outil MJuke (TOOL069) permettant de
*       jouer les musiques converties
*       avec JUKEBOX.
*       (c) 1988  Stephan HADINGER
*
*       Assembleur APW version 1.0
*       tapez "ASSEMBLE MJUKE.SRC"
*       ou "MJUKE.MAKE'
*
        keep  MJUKE
        65816 on
        absaddr on
        mcopy MJUKE.macros
        case  off
        longa on
        longi on
MJCallTable START
                        ; points d'entrée des
                        ; routines
        dc      i4'(TheEnd-MJCallTable)/4'
        dc      i4'MJBootInit-1'
        dc      i4'MJStartUp-1'
        dc      i4'MJShutDown-1'
        dc      i4'MJVersion-1'
        dc      i4'MJReset-1'
        dc      i4'MJStatus-1'
        dc      i4'MJReset-1'
        dc      i4'MJReset-1'
        dc      i4'MJMount-1'
        dc      i4'MJPlay-1'
        dc      i4'MJJuke-1'
        dc      i4'MJCont-1'
        dc      i4'MJStop-1'
        dc      i4'MJWave-1'
        dc      i4'MJNext-1'
        dc      i4'MJSong-1'
        dc      i4'MJTimer-1'
TheEnd  anop
        dc      c'(c) 1988 Stephan HADINGER.'
        END
```

Suite du source en page 44

Wanted

Jean-Yves Bourdin

ToolBox Mag souhaite, entre autres fonctions, servir d'interface entre les utilisateurs et les programmeurs. Utilisateurs, dîtes-nous quels programmes vous aimeriez trouver sur votre GS. Programmeurs, songez que des programmes UTILES mesurent aussi bien vos capacités que des super-démos qu'on regarde une fois et qu'on jette ensuite : curieusement, nous avons constaté, à ToolBox, que nombre de programmeurs GS manquaient d'idées sérieuses de programmes, alors que les demandes des utilisateurs sont claires.

ToolBox Mag souhaite aussi servir de "Centrale" pour les programmeurs : avant de démarrer le travail sur un projet inclus dans cette liste, appelez-nous. Si vous êtes plusieurs sur le même coup, nous vous mettrons en rapport, et vous pourrez collaborer, cela se fait !

Voici donc, dans le désordre le plus absolu, une liste de projets de programmation qui correspondent à une demande des utilisateurs de GS : autrement dit, ToolBox est preneur !

TextEdit : mini... traitement de texte

Avec l'arrivée de TexEdit, le traitement de texte devient presque un jeu d'enfants à programmer. Sauf qu'il n'existe toujours pas de traitement de texte **en français** bâti autour de TextEdit, alors qu'il en existe en anglais. Il faudrait pourtant pouvoir, une bonne fois, jeter à la poubelle ce monstre de GS Write.

Prévoyez deux versions simultanées : NDA et application. Commencez tout de suite de façon modulaire : c'est le genre de programme qui peut toujours grossir.

Le temps que vous écriviez votre code, un nouveau système aura supprimé les bugs de TexEdit du 5.0 –le couper du couper-coller– et en aura introduit d'autres. Préparez-vous à évoluer.

Spooler d'imprimante

La possibilité du spooling est d'origine dans le Print Manager du GS. Précisons que, par Spooler, nous n'entendons pas ici un utilitaire de réseau, mais simplement un logiciel qui permet, sur ImageWriter en premier lieu, de reprendre la main après le début de l'impression, sans attendre l'imprimante.

Il suffit qu'il intercepte les appels à QuickDraw, plutôt que les bitmaps, ce qui ferait des fichiers trop gros, les sauve dans un fichier sur disque, et les renvoie en tâche de fond. Ne pas s'occuper des ports série et du buffer d'impression de la Rom.

A présenter sous forme d'Init, avec un accessoire pour la configuration et la gestion de la queue d'impression, méfiance encore à l'égard des CDEV sur GS. Prévoir peut-être aussi un mini-programme de démo imprimeur de fichiers textes.

Apple Preferred

Aucun de nos programmes de Paint ne gère sérieusement les fichiers graphiques Apple Preferred de plusieurs écrans de large et de plus de deux écrans de long. Or, si vous imprimez une page A4 sur ImageWriter en modes Condensé et Réduction à 50 %, la BitMap correspondante à l'écran fait 1280 pixels de large et plus de 3200 de haut. Et le driver de Laser permet une réduction à 25 % ! Les limites des fichiers Apple Preferred sont de 32768 sur 32768.

Avec Quickie, on peut constituer –et imprimer– de tels fichiers. Avec AppleWorks-GS aussi, mais AppleWorks-GS ne supporte pas le mode 320, plante complètement dans la gestion de ses propres gros fichiers Apple Preferred, et surtout ses fichiers graphiques ne sont pas réellement en bit-map.

Imaginez quels superbes certificats, invitations... on peut faire en gérant une page A4 représentée en BitMap. Votre Paint deviendra un super-Certificate Maker, super-Print Shop, etc. Il faut de la mémoire pour ce genre de logiciel, bien sûr. Et alors ?

On peut aussi rêver d'un logiciel qui supporte les deux modes Draw et Paint, en faisant deux feuilles superposées, des sortes de calques.

NDA Gestion de fichiers

Aucune des petites tentatives que nous avons vues jusqu'ici ne marche vraiment. Et pourtant, ce serait utile et facile à programmer : un NDA qui lirait, afficherait, permettrait les recherches et le copier-coller sur des fichiers textes de Base de données (avec Tabs et Return) issus d'AppleWorks-GS.

Votre agenda en permanence sous la main, les codes postaux, les mnémoniques du 6502, les appels à la ToolBox, et tout ce qu'on peut imaginer. L'essentiel est que l'accessoire ne soit pas dédié à un type de données précis, mais gère tous les fichiers Ascii avec Tabs et Return. Là aussi, il serait bon de prévoir modulaire et évolutif : on peut toujours rajouter des fonctions à ce type de NDA. L'assembleur n'est pas du tout interdit, mais le Pascal et le C s'y prêtent très bien.

Drivers divers

Entre les notes techniques Apple d'un côté, les drivers Epson et ImageWriter à désassembler de l'autre, un programmeur GS sait aujourd'hui écrire un driver d'imprimante. Il peut, en utilisant un mode 25 % impliquant des fontes de taille x4, gérer des imprimantes 300 DPI. AppleWorks-GS permet tout cela sans problème. Or, le prix des imprimantes laser et jet d'encre 300 DPI est en baisse en ce moment...

La traduction, vite...

Un programmeur d'Apple France vient d'écrire un magnifique driver, qui s'appelle "TextPictSaver". Ce driver de "pseudo-imprimante" transforme toute impression en sauvegarde sur disque sous forme de fichier PICT, gérable ensuite dans les programmes de Draw. On peut élargir ou réduire les tracés à volonté. On peut aussi choisir de sauver seulement le texte en Ascii. Magnifique.

Un pépin : c'est un driver pour Mac. Mais une chance : le source en Pascal est fourni avec, et le tout est gratuit ! Il ne reste donc qu'à traduire ce source Mac pour GS. Cela fera un programme super-utile et un excellent exercice de programmation : QuickDraw et les fichiers PICT sont vraiment très voisins sur les deux machines. On gère ces fichiers avec TopDraw/BeagleDraw en 320 et AppleWorks-GS en 640. La course est partie, ToolBox vous donne rendez-vous au poteau.

Tool Soundsmith

N'oublions pas le Tool pour jouer les musiques Soundsmith : entre les notes techniques Apple qui documentent totalement les fichiers Soundsmith et le Tool de Monsieur Hadinger pour Music Studio, tout vous est mâché. Comme démo du Soundsmith Tool, un NDA qui jouerait les musiques Soundsmith en tâche de fond serait le bienvenu.

Taper le carton

La France est coupée en deux pour les jeux de cartes : il y a les adeptes du tarot, et il y a les beloteurs. Les premiers sont comblés par le Tarot du Maître Uhrich. Les seconds se sentent bien orphelins...

CDA calculette de programmeur

S'il y a une calculette de programmeur à côté de votre GS, vous savez de quoi nous parlons : `$6F EOR $32 = ???` La calculette serait aussi bien dans le GS : pour les programmeurs, donc en CDA, mais avec gestion du couper-coller.

Macros souris

MacroMate est bien gentil, mais il a une limite fondamentale : il ne permet pas de "macrotiser" les mouvements et clics de la souris. On peut laisser MacroMate et UltraMacros s'occuper des softs ProDOS 8, et faire un programme de macros qui marche avec AppleWorks-GS.

Césure

Seul Gribouille-GS gère correctement la coupure des mots en français sur GS. Mais ce n'est pas un traitement de textes graphique. Il y a quelque chose à faire, ne serait-ce qu'un accessoire pour AppleWorks-GS...

Note : il n'y a pas non plus de correcteur français sur GS. Franchement, nous n'y croyons pas : dans l'état actuel des technologies logicielles et matérielles, les correcteurs sont à la fois beaucoup trop bêtes et beaucoup trop lents pour être utilisables, même sur Mac fx ; ne vous laissez pas impressionner par les correcteurs et thesaurus anglais dont nous disposons : en français, c'est cent fois plus difficile !

Ecran virtuel

Sur Mac, cela s'appelle "Stepping Out" : un écran virtuel réglable à volonté. Les déplacements sur cet écran virtuel se font en accompagnant la souris, l'écran réel n'étant qu'une fenêtre sur l'écran virtuel. Cela permet d'avoir des fenêtres qui font une vraie page, et de pouvoir tranquillement avoir plusieurs fenêtres ouvertes sur le bureau, tout en voyant toujours ce qu'on fait. Le QuickDraw du GS y est tout aussi prêt que celui du Mac...

Pour les ambitieux

Dans le fond, pourquoi pas un tableur sur GS ? Une gestion de fichiers relationnelle ? AppleWorks-GS est un intégré et, à part lui, il n'y a aucune concurrence !

Idées ToolBox Mag

Si vous voulez des tâches faciles, ToolBox est déjà preneur de deux "suites" au contenu de ce numéro :

- Un NDA qui jouerait des musiques Music Studio en tâche de fond avec le Tool69.
- Une traduction de la routine InitTools de P. Desnoues pour Merlin-16.

■

La disquette 42 Mégas pour IIGS

Hubert Loiseleux

Ca y est, les disques durs à cartouches amovibles Syquest SQ555 sont maintenant importés en France, à des prix du même ordre que les prix US. Donc, cet article à propos du Syquest est écrit sur GS et sauvé sur SQ555.

Mais oui... ça marche !

Il y a en fait bien peu de choses à dire concernant l'utilisation de ces disques durs sur un GS. GS/OS les formate, les partitionne, et les gère sans problème (ProDOS 8 et 16 aussi bien sûr, le Syquest marche aussi sur un //e).

Ils sont très rapides : ils sont donnés à 19 ms, et la vitesse est toujours excellente, quel que soit l'interleave choisi ; inutile donc de s'occuper de l'interleave.

Ils marchent aussi bien avec la carte SCSI Révision C qu'avec la nouvelle SCSI-DMA : ils sont d'ailleurs aussi rapides avec les deux cartes, le DMA n'ajoute pas grand-chose quand le dur est vraiment rapide.

Ils sont munis d'un bouchon SCSI, du câble nécessaire, l'identificateur SCSI se règle très facilement. Bref : branchez, allumez, roulez.

Pour le back-up, le programme d'Apple livré avec la carte SCSI/DMA (Back Up II) marche sans problème (même avec la carte Rom C). Nous recommandons toujours ProSel et/ou Salvation-Renaissance, mais ils sont beaucoup moins indispensables si vous avez deux cartouches (voir ci-dessous).

La solution intelligente

Ce qui est intéressant à expliquer, c'est pourquoi ce type de disques durs à cartouche est, à notre avis, préférable à tous les autres.

① Quand votre disque dur est plein, vous en avez un autre de 42 Mégas pour 800 francs, il suffit de changer la cartouche. Et ainsi de suite, à l'infini.

② C'est la formule rêvée pour les back-ups, permettant d'avoir enfin de vrais back-ups. Quand votre disque dur de 42 Mégas est constitué, faîtes un back-up de la première cartouche sur disquettes 800Ko. Restaurez ensuite ce back-up sur une seconde cartouche, et reformatez les disquettes : vous n'en avez plus besoin.

Vous n'avez plus ensuite, chaque jour, qu'à sauvegarder ou copier sur disquette –ou mieux : sur disque Ram– les fichiers qui ont changé depuis le dernier back-up, puis à restaurer ces fichiers sur la cartouche de back-up. Vous aurez ainsi en permanence deux disques durs strictement identiques et interchangeables. Pensez à nettoyer les "back-up bits" sur les deux cartouches, et à optimiser les deux.

Cela périme les périphériques de back-up sur bande magnétique, qui coûtent au moins aussi cher que le Syquest, et ne peuvent pas servir de disques durs.

③ C'est aussi la formule rêvée pour qui a une machine au bureau et une autre chez lui : transportez votre cartouche, elle tient moins de place qu'un livre.

④ Du coup, cela supprime les problèmes de virus, ou tout simplement ce bazar dément introduit par les collègues/élèves/épouses/copains/neveux etc sur votre dur: votre dur, il est dans votre poche ! Plus de risque non plus qu'on vienne faire en douce quelques copies de vos fichiers secrets en votre absence.

⑤ C'est un investissement intelligent : dans cinq ans, les mémoires de masse se compteront en giga-octets. Le back-up sur disquette 800Ko sera impossible en pratique. Mais sur cartouches 42 Mégas, cela sera tout-à-fait praticable.

On trouve déjà de premiers logiciels sur cartouche Syquest : la cartouche devient un standard.

⑥ Tous ceux qui ont des données à archiver ou à transmettre en volume important –quel que soit leur système d'exploitation– ont enfin ce qu'ils cherchaient. A ce propos, combien avez-vous de boîtes de disquettes pleines de "divers" ? Ne serait-il pas malin de regrouper tout cela dans une seule cartouche ?

⑦ C'est peut-être le point décisif : le Syquest SQ 555 est un vrai SCSI. Cela signifie qu'il est compatible avec tout micro. Pas seulement le GS, mais aussi le Mac, et même –excusez ces références inconvenantes– les IBM ou les Atari ; munis de cartes SCSI bien sûr.

Avouez-le : le GS n'est pas le seul micro que vous fréquentez (hélas...). Avouons-le aussi : il y a un Mac chez Toolbox. Ce Mac partage avec les GS ses imprimantes, mais il partage aussi son dur. Il suffit de le débrancher du GS pour le brancher sur le Mac, et de changer la cartouche, en 20 secondes maximum. Exactement la même chose, dans le fond, que le lecteur 800Ko.

C'est là que se manifeste la vraie force de la notion de standard : pensez seulement à étiqueter vos cartouches correctement, et vous avez votre bibliothèque prête pour GS, Mac, IBM, Atari, etc. Vive le SCSI.

A savoir

Quelques points à savoir si vous passez au Syquest :

① Sur Mac, si le système reconnaît parfaitement le Syquest, l'application Apple File Exchange dans sa version du système 6 d'Apple est affectée d'un bug gênant. Cette application, chargée de gérer les conversions entre disquettes et systèmes différents (MS/Dos, ProDOS, Mac/OS), perd les pédales devant cette "disquette dure" SCSI 42 Mégas éjectable, même quand Apple File Exchange a précisément été lancé depuis cette disquette !

Seuls les vieux routiers du GS savent se tirer de ce bug Mac : en utilisant l'ancienne version d'Apple File Exchange spéciale pour GS, appelée "PassPort" ; la disquette Mac livrée avec les classeurs GS, vous vous souvenez ? Il ne faut RIEN jeter...

② Le câble livré avec le Syquest permet de chaîner plusieurs engins SCSI, mais à condition de mettre le Syquest en fin de chaîne. Si vous voulez chaîner deux Syquest l'un derrière l'autre, il faut demander un câble spécial (pas cher) dit "SCSI/SCSI".

③ Le ventilateur s'entend un peu, et vous ne pourrez pas mettre le tout à la cave, parce que vous aurez des échanges de cartouche à faire. En pratique, le bruit n'est pas gênant.

④ Le Syquest permet d'échanger les cartouches "à la volée" exactement comme on fait d'une disquette 800Ko, le fait de sortir la cartouche éteint automatiquement le moteur. Nous ne le conseillons cependant pas : les têtes étant fragiles, nous conseillons d'échanger la cartouche seulement après avoir éteint le lecteur. Il est aussi conseillé, avant d'éteindre le lecteur, de ranger les têtes : le Finder (commande Eteindre) ou ProSel-16 (commande Cold ShutDown) s'en chargent très bien.

Notre conseil

Si vous n'avez pas encore de disque dur : vous êtes gagnant. La patience est parfois récompensée...

Si vous en avez un, regardez ce vieux machin :

- Il est définitivement limité à ses 20 ou 40 Mégas. Si vous voulez plus, il faut rajouter un autre dur !
- Il n'est pratiquement pas transportable, même s'il est externe.
- S'il est possible de l'utiliser sur d'autres machines, il faut faire à chaque fois un back-up, reformater, etc. Curieuse transportabilité...
- Tout le monde y met son grain de sel, sans parler des virus.
- Les back-ups sur disquettes, ce n'est pas vraiment la joie...
- Dans quelques années, il sera ce qu'est aujourd'hui le Profile 5 Mégas.

Alors songez qu'il existe des gens qui ne lisent pas ToolBox Mag (surtout son premier numéro) : ils ont certes bien tort, mais ne serait-ce pas leur rendre service que de leur faire découvrir les joies du disque dur d'occasion, pendant que vous goûtez à celles de l'amovible neuf ?

Un problème de maintenance ?
Une solution :
ToolBox
Téléphonez au (1) 30 76 18 64

GS News

Eric Weyland

HyperStudio 2.1 (Roger Wagner)

Encore une nouvelle version pour HyperStudio ; la version 2.1 donne maintenant la possibilité de distribuer vos piles sans pour autant obliger l'utilisateur final à posséder HyperStudio.

Il est à noter que les programmeurs d'HyperStudio ne sont pas effrayés par la concurrence que va apporter HyperCard GS. Ils ont notamment déclaré que "si Apple réalise un logiciel hypermédia de médiocre qualité, cela renforcera HyperStudio d'autant plus."

Duet (Cirtech)

Voici dans son intégralité le communiqué de presse de ToolBox de fin Juillet concernant la carte Duet de Cirtech. Il y a fort à parier que l'Apple IIGS va devenir le compatible Macintosh II le moins cher du marché.

«ToolBox est heureux d'annoncer une innovation majeure en matière de produits Apple :

Duet de Cirtech

Duet transforme l'Apple IIGS en un Macintosh couleur puissant et bon marché !

Pour la première fois, avec Duet, il devient possible de faire tourner logiciels Mac et GS sur le même ordinateur.

Duet comprend une carte co-processeur (qui s'enfiche dans tout port standard du GS) et un logiciel de liaison.

La carte est munie d'un processeur 68020 à 16 Mhz, d'un Méga de Ram extensible à huit Mégas sur la carte elle-même, et d'une Rom Cirtech. Les périphériques de l'Apple IIGS (clavier ADB, lecteurs de disques, etc.) sont dès le départ compatibles Macintosh : ils sont donc reconnus de façon totalement transparente par les applications Mac.

Une innovation majeure de cette carte est que le processeur 68020 et le processeur 65816 du GS opèrent ensemble : le 65816 est utilisé pour gérer les entrées-sorties, libérant ainsi le 68020 pour faire tourner les logiciels et le système du Mac. Nul n'ignore que dans un Macintosh "normal", le processeur doit gérer lui-même l'essentiel des entrées-sorties, ce qui diminue sa performance d'ensemble. Le système du Duet lui permet de dépasser en vitesse le Mac IIcx !

Les options

Le Duet est livré en standard avec 1 Méga de Ram, extensible à 8 Mégas sur la carte Duet.

Un co-processeur arithmétique 68882 pourra être inséré dans un support prévu sur la carte.

La disponibilité

Le Duet est actuellement en phase finale de développement chez Cirtech-Ecosse. Sa disponibilité commerciale est prévue pour Décembre 90.

La commercialisation

Le Duet sera commercialisé en France par ToolBox, correspondant Cirtech sur le continent. Le Duet confirme ce qui est par nature la vocation de l'Apple IIGS : seul ordinateur personnel capable de faire tourner à la fois les logiciels Apple //, MS-Dos et Macintosh, il est l'ordinateur le plus polyvalent du marché.»

Directeur Apple II

C'est le titre d'un des nouveaux dirigeants d'Apple-USA, Ralph Russo. Ce poste a été créé à la direction d'Apple pour relancer sérieusement l'Apple II, et tout particulièrement le GS. Il a rassemblé autour de lui une petite équipe technico-commerciale qui en veut...

Driver Scanner Apple High Speed SCSI Card

La carte SCSI HIGH SPEED de chez Apple Computer est maintenant livrée avec un driver depuis longtemps attendu : un driver du scanner Apple. Espérons que les programmeurs du côté de Mariani Avenue à Cupertino ont pensé à ce driver lors de la conception d'HyperCard GS.

ToolBox Reference Volume 3

Le troisième tome du ToolBox Reference est disponible dans sa version définitive. Il contient tous les aspects de la ToolBox qui ont évolué depuis l'apparition des ressources dans l'univers de l'Apple IIGS. L'ouvrage contient plus de 1000 pages d'informations de référence.

Apple IIGS Firmware Reference Update Roms 03

La dernière édition du Firmware Reference est maintenant disponible. L'ouvrage décrit le Firmware de l'Apple IIGS Roms 03. Après la lecture de cet ouvrage, il est maintenant évident que les développeurs n'ont pas besoin d'arguments supplémentaires contre la protection anti-copie. L'upgrade entre l'Apple IIGS Roms 01 vers les Roms 03 ont dû leurs donner toutes les raisons d'abandonner ce genre de protections qui, en fait, pénalise l'utilisateur honnête.

Mercury (MECC)

Mercury, un traitement de texte pour Apple IIGS de très bonne facture est maintenant disponible. Il est commercialisé à un prix très abordable.

Nous avons apprécié en particulier la facilité de l'intégration des images dans le texte et de la gestion de ces images avec ses deux "modes" graphique et texte, et sa gestion intelligente de la mémoire. A notre avis, il est largement supérieur à BeagleWrite (ex-MultiScribe GS), et nous ne dirons rien de GS Write...

Designer Prints (MECC)

Ce logiciel est le genre de programme qui a beaucoup d'avenir sur le GS : entièrement standard GS, il ne fait qu'une chose, et il la fait bien.

Cette chose, en l'occurrence, ce sont des affiches : vous composez votre page avec tout un tas d'outils graphiques type objet (très belle collection de bordures par exemple), puis vous l'imprimez avec un agrandissement allant d'un facteur 1 à un facteur 10. Designer Prints se charge d'imprimer les unes à la suite des autres des pages qui sont des fragments du dessin agrandi, en laissant les marges nécessaires. Vous n'avez plus qu'à coller les feuilles pour faire des affiches de plus d'un mètre ! Ce programme gère parfaitement la LaserWriter comme l'ImageWriter (en couleurs pour cette dernière). Si vous passez à ToolBox un de ces jours, regardez bien les murs...

Les lecteurs de **Programming The Apple IIGS In Assembly Language** de **Eyes and Lichty** peuvent recevoir une feuille d'errata en écrivant à :

Ron Lichty – P.O. Box 27262 – San Francisco
CA 94127 – USA.

Mise en garde

Utilisateurs d'Apple IIGS, attention : un programme d'infection des disquettes GS a été récemment mis en circulation par quelques pirataillons de banlieue. Il s'appelle "Spy Format Expert".

D'apparence anodine, il donne la possibilité de formater une disquette 3,5", en simple ou double face, avec l'interleaving de votre choix ; jusqu'à présent rien de bien original... Les autres options proposées sont par contre plus pernicieuses et beaucoup plus dangereuses.

En effet, il donne non seulement la possibilité de choisir la valeur des marqueurs physiques des blocs d'adresse et de données, ainsi que la valeur des octets de synchronisation, mais aussi le choix de placer un virus sur le bloc de boot de la disquette : Load Runner, Odyssey, etc...

Nous prévenons charitablement les auteurs de ce 'programme' qu'ils feraient mieux de se consacrer à de la programmation plus constructive ; la programmation d'une version GS de Conan est certainement beaucoup plus difficile... même les soit-disant 'Jésus de l'informatique' peuvent être facilement retrouvés par le service juridique d'Apple...

Aux utilisateurs, nous conseillons tout simplement de jeter ce programme si jamais il leur parvient: Spy Format Expert n'est pas disponible par l'intermédiaire de ToolBox...

Les mêmes peuvent vous envoyer leur disquette d'exemples en format Merlin 16, et plus seulement APW.

Cours De Pascal (ByteWorks)

Si vous voulez vous mettre à programmer sérieusement, de façon "pro", votre GS, en-dehors de l'Assembleur vous n'avez aujourd'hui que deux choix : le Pascal et le C.

Pour le C, vous l'apprendrez avec François Uhrich dans ToolBox-Mag et votre Kernigan et Ritchie.
Pour le Pascal, ByteWorks propose une excellente affaire : un

"package" comprenant Orca/Pascal (excellent Pascal intégré à APW/Orca) et un excellent cours "par la pratique" d'initiation au Pascal (un vrai cours : exercices, devoirs, et tout : les vacances sont finies).

Très bel ensemble, qui fera de vous un programmeur respecté sur GS.

Soft Quickie (Vitesse)

La version 2.01 du logiciel GS pour le scanner à main Quickie est disponible. Le programme a été entiérement ré-écrit. Il travaille maintenant en mode graphique 320 x 200 ou en 640 x 200. Il est capable de travailler avec des nuances de gris, et dispose d'une très grande quantité de réglages permettant d'obtenir une qualité de digitalisation tout à fait exceptionnelle.

Le programme se présente sous deux formes : un programme S16 et un NDA. Avec le NDA, il est possible de faire une digitalisation sans sortir de son application. Par exemple, on peut utiliser Quickie sans sortir du module de PAO d'AppleWorks GS.

Il est recommandé de posséder quelques Méga-octets de mémoire pour avoir la possibilité de digitaliser des dessins de taille importante ; plus vous avez de Ram, plus la taille de vos digitalisations pourra être importante.

Il est à noter que le logiciel Quickie est le seul à ce jour à gérer correctement les fichiers graphiques Apple Prefered. Nous avons particulièrement apprécié la vitesse du Quickie dans le processus de digitalisation lui même (moins de dix secondes) et la grande qualité du programme dans les agrandissements ou les réductions des images ou partie d'image.

Les utilisateurs d'Apple //e n'ont pas été oubliés ; Quickie peut maintenant fonctionner en utilisant les résolutions graphiques de cet ordinateur.

Pour vous montrer la valeur du Quickie, nous avons mis sur la disquette quelques dessins digitalisés avec Quickie : n'oubliez pas que cela se fait en 10 secondes !

Nous attendons maintenant avec impatience le logiciel permettant la reconnaissance des caractères, l'O.C.R. (voir les "fumées sans feu" de ce numéro).

CDrive (Ingenuity)

Dave Westbrook de la société Ingenuity (ex Applied Ingenuity) a présenté à l'AppleFest / ComputerFest le CDrive, un disque dur interne pour l'Apple //c et l'Apple //e. Bientôt disponible en France.

Turborez

Cette carte permet de doubler la résolution graphique de l'Apple IIGS. A la dernière AppleFest / ComputerFest, elle a fait grande impression parmi les revendeurs Commodore.

Dommage qu'elle ne soit pas compatible avec QuickDraw...

Logo Writer GS (Logo Computer System)

La sympathique tortue si chère aux programmeurs en herbe, arrive enfin sur Apple IIGS. Logo Writer est un véritable environnement de programmation sous GS/OS comprenant un langage (le Logo) et un véritable traitement de texte pour l'édition des programmes. Il est possible d'animer des images, de réaliser des effets sonores, et de faire de la musique.

L'enfant grâce à ces outils va rapidement maîtriser des concepts mathématiques importants comme les angles, les distances, les formes... Logo Writer existe en trois versions ; une version pour les enfants de moins de trois ans, une autre pour ceux ayant entre 4 et 8 ans, et une dernière version pour les plus de 8 ans. Avec cette dernière version, il est possible d'utiliser le système Lego qui comprend des briques, des engrenages, des moteurs... faisant ainsi de Logo Writer un outil de premier ordre pour l'apprentissage des principes de la physique.

Nexus (Golem Computers)

Les boutons et les piles sont vraiment très en vogue depuis quelques mois. Pourtant, Nexus n'est pas un hypermédia semblable à tous les autres. Nexus permet de connecter (créer des liens) entre des fichiers texte et des graphiques.

Il est par exemple possible de prendre un fichier créé avec AppleWorks et de lier un mot clé à l'intérieur de ce document vers un autre fichier texte. Dans un document décrivant les joies du surf, il est possible de connecter le mot 'planche' sur un fichier qui décrit plus précisément l'utilisation de cette 'planche'.

Dans un document Nexus, on peut consulter tous les liens qui existent dans ce document. Il est bien entendu possible d'ajouter d'autres liens et d'en supprimer. Nexus fonctionne sous GS/OS ; il est sous mode desktop, souris, menus déroulants, etc. et toutes les commandes ont des équivalents clavier. Pour les liens graphiques, il suffit tout simplement de définir une zone sur ce graphique, où l'utilisateur devra cliquer ; à partir de là, vous pouvez afficher une autre image ou charger un fichier texte.

Nexus a été écrit par Tracy Valleau ; il nécessite le système d'exploitation GS/OS 5.0.2 et 1.25 Méga-octets de Ram. C'est un très bon moyen d'aborder le monde de l'hypermédia.

Your IIGS Guide (CompuShare)

Ce guide (en américain) contient une véritable mine de renseignements, de trucs, de conseils, d'adresses, pour vous aider à maîtriser votre Apple IIGS. Le guide décrit tous les périphériques de l'Apple IIGS ; il évoque les extensions possibles de la machine tant au niveau matériel que logiciel. D'après l'éditeur, il a été conçu pour aider le consommateur à acheter ce dont il a besoin en lui évitant toute dépense superflue. Il est disponible chez :

CompuShare – 1800 North Road – Bothell
WA 98012-6217 – USA.

Call Box Programmers Association

Vous êtes un fanatique de CallBox ? So What Software, l'éditeur de l'outil de programmation CallBox, vient de créer l'Association des programmeurs sous CallBox (CallBox Programmers Association - CBPA). Pour $25, l'Association founit un support technique sur le système de programmation CallBox : un fanzine trimestriel, des notes techniques, des disquettes d'exemples de programmes sous CallBox, des utilitaires, et des programmes réalisés grâce à CallBox. L'auteur de CallBox, Bill Stevens, a déclaré que le premier envoi aux adhérents de l'Association comprenait entre autre 12 notes techniques décrivant l'utilisation des appels GS/OS de classe 1 à partir du Basic, les techniques à utiliser pour réserver des blocs de mémoire ; un disque d'exemples contenant des templates pour la version 5.0.2 de GS/OS, sur la programmation du son, sur la programmation en mode desktop ... Pour plus de renseignements, envoyez votre carte de garantie chez CallBox, ou prenez contact avec :

CallBox Programmers Association
C/O So What SoftWare – 10221 Slater Avenue
Suite 103, Fountain Valley, CA 92708, USA.

Ram 4000 (Chinook)

Cette nouvelle carte d'extension mémoire permet de porter la mémoire adressable de votre Apple IIGS jusqu'à 4 Méga-octets. Elle utilise des Ram chips standards de 1 Méga-byte. La carte peut être étendue à 1, 2, ou 4 Méga-octets. La Ram 4000 suit les recommandations d'Apple. Cela signifie qu'elle fonctionne avec des GS Rom 01 ou 03, et qu'elle est compatible avec la nouvelle carte SCSI Apple (High Speed SCSI Card).

GS Sauce (Harris Laboratories)

Deux originalités de cette carte : entièrement en puces CMos, elle consomme peu. Ensuite, elle accepte les puces SIMM 1 Mégabit du Mac, dont le prix est en chute libre en France. Elle aussi suit les recommandations Apple, et peut être étendue à 1, 2, ou 4 Mégas.

Once Upon A Time II (Compu Teach)

Demandez aux très jeunes enfants de choisir entre faire des dessins et lire des histoires ; vous pouvez alors attendre la réponse pendant longtemps... Once Upon a Time II est un programme éducatif qui incorpore les deux activités. En quelques minutes, les enfants peuvent concevoir des pages illustrées, écrire du texte sur ces images, puis relier les pages pour fabriquer un livre. Ils peuvent alors relire cette histoire sur l'écran, et l'imprimer en couleur ou en noir et blanc.

High Speed SCSI Card (Apple Computer)

La nouvelle carte SCSI d'Apple Computer permet des opérations d'Entrée/Sortie entre un Apple II et un périphérique tel un disque dur ou un lecteur de CD Rom 10 fois plus rapides.

Sans augmentation de prix la High Speed SCSI Card est le moyen le plus rapide pour transférer des informations entre un Apple II et un périphérique SCSI. Attention, il n'y a vraiment amélioration des performances que lors des traitements de très gros fichiers ; plus le fichier a une taille importante, plus l'accélération est manifeste.

Ces performances sont essentiellement dues à une nouvelle caractéristique dans le transfert des données, le DMA (Direct Memory Access). Cela signifie que les données sont directement transmises entre la mémoire de l'ordinateur et le périphérique SCSI, avec une utilisation minimum du micro-processeur de l'Apple II.

Cette nouvelle carte transfère les données à un taux de 1 Méga-octets par seconde sur Apple IIGS, et 500Ko par seconde sur un Apple //e. En plus du DMA, cette nouvelle carte intègre une révision du SCSI Manager et d'un nouveau firmware qui contribuent aussi à l'augmentation de vitesse.

Katie's Farm (Lawrence Productions)

Transformez votre Apple IIGS en basse cour... Ecoutez les poules, le cri aigü des oies... Goûtez les framboises ; hum hum, elles sont délicieuses. Attrapez ces adorables poussins pendant que Katie ramasse les oeufs. Regardez Tante Thelma

traire la vache dans l'étable... Katie's Farm est le second programme de Lawrence Production (le premier étant Mc Gee). Il s'agit de jeux entièrement graphique, sans aucun texte, jouables à la souris et, agrémentés d'animations et d'effets sonores digitalisés. La qualité de ces jeux est exceptionnelle. Ils permettent aux jeunes enfants de découvrir et d'explorer un univers tout à fait fascinant. BroderBund Software vient de signer un contrat de distribution pour ces deux titres.

Second Chance Version 2.0 (Raptor Inc.)

Ce programme est composé de 19 routines graphiques permettant d'améliorer l'aspect des images digitalisées, ou des dessins qui ont été convertis à partir d'un autre ordinateur, et qui sont souvent de moins bonne qualité après conversion.

Color + et SuperGraphix II (El Kassir SoftWare)

Color + est un programme de dessin qui permet de mixer les modes graphiques 320 et 640 en utilisant jusqu'à 256 couleurs. Le logiciel est fourni avec un programme de mixage des couleurs, un programme de récupération de shapes, des éditeurs de fontes, motifs et palettes de couleurs, ainsi que des programmes de démonstration.

Supergraphix II est un utilitaire améliorant le Basic AppleSoft par la technique d'ajout de commandes par l'Ampersand (&). Le programme permet de travailler en mode graphique 320 et 640 avec 256 couleurs. Il permet de travailler en SHGR à partir de l'AppleSoft.

A Lire

Une nouvelle série de livres traitant des réseaux, destinés à l'utilisateur, aux programmeurs et aux développeurs vient de sortir chez Addison Wesley. Ils décrivent les opérations de base ainsi que les techniques les plus avancées.

Les titres pour le moment disponibles sont : **UnderStanding Computer NetWorks** (96 pages), il s'agit d'une introduction aux concepts de base des réseaux ; **AppleTalk NetWorks System** Overview (200 pages), l'ouvrage décrit les composantes d'un réseau Apple et explique les techniques employées pour réunir dans un réseau ces différentes composantes ; **Inside AppleTalk** (576 pages), est un ouvrage technique sur le réseau AppleTalk. Ces trois ouvrages sont respectivement des livres d'Introduction, de Référence , et d'Approfondissement technique.

AppleFest (San Fransisco)

Cette année, l'exposition n'aura pas lieu ni à la date ni à l'endroit habituels. En effet, elle se tiendra au Long Beach Convention Center à Long Beach en Californie les 7 et 8 décembre 1990. Apple aura un stand à l'AppleFest.

Chinook SCSI Utilities (Chinook)

Une version sous GS/OS des utilitaires Chinook est disponible. Elle travaille aussi bien avec la carte SCSI Rev C que la carte SCSI High Speed.

Halls Of MonteZuma (SSG)

Enfin un Wargame sur GS. Le jeu nous vient d'Australie. Ian Trout, l'un des programmeurs de l'équipe, n'hésite pas à prendre les commandes par téléphone tout en gardant sa bonne humeur... Normal me direz vous ? un petit détail, la conversation téléphonique avait lieu à 3 heures trente du matin heure locale... ToolBox pourrait quand même s'offrir un Psion pour le calcul des décalages horaires avec les antipodes... Il est vrai que nous étions tous pressés de se faire une petite guerre.

SSG édite aussi Reach for the Stars sur l'Apple IIGS. Halls of Montezuma est un jeu en couleur, entièrement graphique, en mode desktop et sous GS/OS 5.02. Il est au standard ProDOS. Plusieurs scénari sont proposés : Mexico City, Belleau Wood, Iwo Jima, Okinawa - the Shuri Line, Okinawa ; the final redoubt, Pusan, Inchon, Hue.

L'action débute en 1848 par la prise de Mexico City, et se poursuit jusqu'aux deux guerres mondiales, puis les guerres de Corée et du Vietnam. Vous commandez le prestigieux corps des Marines.

Attention, l'adversaire quand il est commandé par l'Apple IIGS est particulièrement redoutable ; il ne vous fera pas de cadeau. Il n'est pas facile de changer le sens de l'histoire. Tout doit être pris en compte : déplacements, ravitaillements, combats, soutien aérien...

Le programme est fourni avec un module Warplan permettant la création de ces propres scénari ; on peut aussi s'en servir pour modifier ceux existant pour créer des configurations historiques un peu différentes. Pour les tester, vous avez la possibilité de faire jouer l'ordinateur contre lui même.

Blue Angel (Accolade)

Vous appartenez à une escadrille de vol en formation. Le jeu se présente sous la forme d'un simulateur de vol. Bien entendu, il faut surveiller de près les évolutions des autres

avions de l'escadrille. Pour le moment nous n'avons vu qu'une pré-version du jeu. Bientôt disponible...

Centauri Alliance (Broderbund)

Décidément, Broderbund nous surprendra toujours par les qualités d'animation graphique qu'il est capable de tirer de 64Ko et du mode HGR de l'Apple II : Centauri Alliance est bel et bien au jeu de rôles ce que "Prince of Persia" est au jeu d'arcade.

Jeu de rôles type "Space Opera" (la Science-Fiction des années 50), il est néanmoins capable de reprendre vos équipes de Bard'sTale, Ultima, Wizardry, etc.

Vous vous déplacez dans un Labyrinthe en trois dimensions type Dungeon Master. La carte du jeu s'affiche en permanence sur l'écran (plus besoin de papier quadrillé). Les combats ont lieu sur une grille en trois dimensions où ennemis et obstacles sont représentés par des hexagones. Il se pilote à la souris et au joystick. C'est donc un jeu de rôles moderne, d'aujourd'hui.

Le point essentiel est cependant celui-ci : le jeu est intéressant ! Bravo Broderbund.

Souris Stylo

Après la souris sans queue à infra-rouges, après la souris immobile ventre en l'air, voici la souris-stylo, de IMCS Inc : c'est la magie du port ADB du GS et du Mac. Excellente pour dessiner avec précision, pour tracer sa signature par exemple.

En-dehors des tablettes graphiques type Kurta, elle n'a qu'un concurrent pour les tracés : le crayon sans souris et le Quickie.

Guide Apple II Troisième Edition

Rappelons d'abord que ce guide est gratuit et qu'il devrait être disponible auprès de votre concessionnaire Apple. Vous pouvez aussi le demander directement chez Apple France en téléphonant au (1) 69 86 37 00, ou par Minitel sur le 3616 code APPLEA. D'après Apple, la dernière édition a été distribuée à 13000 exemplaires.

ToolBox est toujours ignoré par le guide... De plus, le livre de programmation édité et diffusé par ToolBox "le IIGS Epluché" est mentionné dans le guide sous un faux titre "Documentation technique", et la diffusion en est faussement attribuée... à la société Bréjoux ! Nous attendons avec impatience le rectificatif qui s'impose dans la prochaine édition du guide.

A part cela, notre technicien de maintenance a lu avec intérêt la partie consacrée au diagnostic des pannes éventuelles sur votre Apple, qu'il soit //e, //c ou IIGS. Il vous conseille de garder les emballages de vos ordinateurs et périphériques, car les 80 pages de précieux conseils préconisent le plus souvent de ramener le matériel en panne chez un concessionnaire agréé Apple...

Ampersand

D'ici la fin de l'année ce sympathique fanzine du sud de la France va devenir gratuit et avoir une diffusion nationale.

Nouveautés Francaises (ToolBox)

Ce n'est pas moins de quatre nouveaux logiciels que ToolBox présentera à Apple Expo. Ils ont tous les quatre été programmés par des Français, comme quoi qu'il n'y a pas qu'aux Etats-Unis que l'on apprécie le IIGS. Après le IIGS Epluché et Tarot cela prouve bien qu'il existe en France des gens aimant leur machine et prêts à lui consacrer beucoup de temps. Aux utilisateurs de démontrer maintenant qu'il existe un marché pour le logiciel Apple IIGS. Vous trouverez des informations sur ces quatre programmes dans les lignes qui suivent.

Les fanatiques du GS ne parlant pas notre langue n'ont pas été oubliés car Bouncing Bluster II, Photonix II, et Space Shark ont aussi été pensés pour les utilisateurs anglo-saxons. Amis américains ne renoncez pas pour autant à l'étude de la langue française, mais vous aurez bientôt la possibilité de jouer au Tarot sur votre GS et de dévorer le IIGS Epluché...

Bouncing Bluster II (ToolBox)

Voici un grand classique du jeu puisqu'il reprend le principe très connu du casse-briques. Le jeu a été programmé en assembleur sous APW par deux jeunes Français de la région parisienne Jean François Doué et Jean Michel Vallat, tous deux étudiants. Bouncing Bluster a été tout d'abord diffusé en version ShareWare tant en France qu'aux Etats-Unis ; la revue A+ Incider l'a d'ailleurs présenté comme le meilleur casse-briques sur toute machine confondue... La version II apporte des améliorations notables à la version ShareWare. Tout d'abord le jeu tourne maintenant sous système 5.0. De la musique et des effets sonores variés accompagnent maintenant le joueur dans sa difficile mission... Il a été débuggé et de nombreuses 'surprises' ont été ajoutées. Le jeu est accompagné d'un éditeur de tableaux très complet qui permet de choisir jusqu'à la capsule qui tombe du ciel lorsque l'on détruit une brique. Toutes les personnes ayant acquitté les

droits du ShareWare recevront gratuitement la version II de Bouncing Bluster. Bouncing Bluster II est au standard ProDOS ; il n'est donc pas protégé contre la copie.

G F P : Gestion Familiale personnelle (ToolBox)

Voici le premier programme de comptabilité personnelle en français pour Apple IIGS. Comptabilité Privée va tenir de très près votre compte en banque. C'est l'équivalent de Budget Familial de l'Apple IIe. Comptabilité Privée a été écrit par un fidèle de ToolBox (devinez qui ?) en assembleur sous APW. Il est entièrement conçu pour tourner sous GS/OS 5.0. Il n'est absolument pas protégé contre la copie, mais je suis sûr que personne ne risquera à le diffuser illégalement vu le mode de protection utilisé...

Photonix II (ToolBox)

Le copieur le plus original et le plus rapide pour reproduire des disquettes Apple IIGS et Macintosh. Tout le génie de ce programme est dans son interface et sa présentation. Photonix II est lançable à partir du Finder ou de ProSel 16 et peut être copié sans difficultés sur un disque dur. Il détecte automatiquement les virus susceptibles d'infecter vos disques au moment de la lecture et vous propose de les supprimer. Photonix II fonctionne parfaitement avec les IIGS équipés de Roms 03. De nombreuses surprises parsèment le copieur... Deux reproches cependant : Photonix II oblige à rebooter au moment du Quit (commande Q), il est vrai que cela ne prend que quelques secondes sur un dur ; il ne fonctionne pas sous AppleTalk, configuration rare mais défaut gênant pour ceux qui fonctionnent en réseau, ceux de ToolBox par exemple...

Les auteurs de Photonix II sont les créateurs de Nucleus et de Modulae deux démonstrations montrant jusqu'où on peut aller lorsque que l'on aime sa machine. Nous attendons avec grande impatience les programmes à venir de O. Goguel et de P. R. Richard...

Space Shark (ToolBox)

Encore un programme écrit par une équipe de français, de l'Est de la France cette fois ci. Il s'agit d'un jeu d'arcade genre Defender qui semble bien être le plus rapide qui ait jamais été écrit pour l'Apple IIGS.

Space Shark comprend quinze niveaux de jeu dont la difficulté est progressive. Dans le premier niveau, le joueur est confronté à deux espèces différentes d'aliens. A chaque nouveau niveau qu'il atteindra, le joueur rencontrera une nouvelle espèce d'alien dont, bien sûr, l'agressivité augmentera. Le jeu se joue à la souris ou au joystick. Les effets sonores sont spectaculaires, surtout pour les GS équipés d'une carte stéréo.

Block Out (California Dreams)

Attention voici un jeu d'un type tout à fait nouveau... Asseyez vous confortablement avant de commencer à jouer. Vous risquez de ne pas décoller de votre écran pendant un bon moment, une fois que vous serez accroché, vous ne pourrez plus vous passer de ces maudits blocs colorés.

Au départ, sur l'écran, il y a une fosse représentant un espace en trois dimensions ; la longueur, la largeur, et la profondeur de cet espace sont configurables. Le jeu consiste à manoeuvrer chacun des blocs qui se présente dans une position correcte afin de former des couches. A chaque fois qu'une couche est constituée, elle disparaît, dégageant ainsi de la place, et vous faisant gagner des points.

Blockout va développer vos réflexes et votre esprit de réflexion. Le jeu va tester la coordination de vos mouvements et la vitesse de votre pensée. Parmi les caractéristiques du jeu, il existe près de 1000 combinaisons possibles d'espaces et de positionnement des blocs ; il est bien entendu possible de sauvegarder une partie et de s'entraîner avec le mode Practice.

Primo (Cirtech)

Voici une nouvelle carte mémoire pour Apple IIGS avec quelques petites particularités très intéressantes. En effet la carte Primo est la première carte multifonctions pour Apple IIGS. Elle s'installe dans le slot d'extension mémoire de votre GS, Primo est extensible à 4 Méga-octets, et parce qu'elle est entièrement compatible DMA, elle cohabite sans aucun problème avec le nouveau GS et la carte SCSI High speed.

De plus, Primo donne la possibilité d'adjonction d'un disque dur interne ou d'un port SCSI. Le disque dur interne est disponible en 20 ou 40 Méga-octets et s'enfiche facilement dans le port d'extension de la Primo. C'est un disque dur aux temps d'accès très corrects ; il est reconnu comme Rom disque par le système. Cela est idéal pour les nababs n'ayant plus de slots disponibles... On peut aussi adjoindre un port SCSI à la carte PRIMO ; il permet de connecter 8 périphériques externes à la norme SCSI. Bien entendu tout cela est compatible avec GS/OS, ProDOS 8, CP/M Plus, Pascal et DOS 3.3.

Théorie Du Bordel Ambiant (Roland Moreno - Editions Belfond)

Un livre d'un fanatique d'Apple cela se dévore. C'est le cas de TBA qui, en 300 pages, va vous révolutionner votre façon de

raisonner et vous offrir des horizons nouveaux sur les applications et l'avenir de l'informatique, entre autre domaine abordé. Beaucoup d'idées de SoftWares inexistants sont présentées : l'exemple d'un logiciel futuriste qui à partir d'un téléscripteur d'une agence de presse fabriquerait les images de synthèse d'un animateur TV présentant le journal de 20 heures est particulièrement amusant.

Genesys (Simple SoftWare Systems International)

Genesys va pouvoir vous faire gagner de nombreuses semaines de travail si vous programmez des applications de bureau sous GS/OS. Il permet de définir graphiquement tous vos menus, fenêtres, boîtes de dialogue, etc... au lieu de travailler octet par octet dans un éditeur de texte.

Sans faire de compilation ou de linkage, vous voyez immédiatement les résultats de ce que vous êtes en train de faire sur l'écran. Une fois que l'interface utilisateur de votre programme est créée, Genesys va automatiquement fabriquer un code source entièrement commenté dans le langage de programmation de votre choix (*RESSOURCE INFO REPORT, APW/ORCA 65816, APW C, APW REZ, MERLIN 16, MICOL BASIC, MPW IIGS ASSEMBLER, MPW IIGS C, MPW IIGS PASCAL, ORCA C, ORCA PASCAL, TML PASCAL II*).

Les programmeurs débutants peuvent étudier les sources ainsi constitués pour comprendre la programmation en mode desktop. Les non programmeurs peuvent utiliser Genesys pour adapter les programmes utilisant des ressources. Les utilisations de Genesys sont infinies : renommer les titres des menus, changer la couleur des fenêtres, ajouter des équivalents clavier...

Garanties

Tous les utilisateurs d'Apple peuvent savourer l'extraordinaire morceau de littérature à la Pierre Dac que constitue le texte intitulé "Limitation de Garantie et de Responsabilité", qu'on trouve à la page 2 de couverture des manuels Apple.

Si Apple a le droit d'écrire de telles énormités, ToolBox Mag ne va pas se gêner : nous aussi, donc, nous vendons tout ça "en l'état", l'acheteur prend tous les risques, etc.

Simplement, en cas de pépin, nous vous demandons d'en informer ToolBox Mag. Car nous prenons un engagement supplémentaire : celui de rectifier dans un numéro suivant de ToolBox Mag les erreurs qui nous auront été signalées. Merci.

Where In The USA Is Carmen San Diego (Broderbund)

Encore une mission difficile en perspective... Carmen San Diego vient de s'évader de prison où elle était recluse en Europe. Elle a pour objectif les Etats-Unis. Dans cette enquête vous découvrirez la géographie des Etats-Unis à mesure que vous voyagerez dans tout le pays sur les traces de la redoutable criminelle. Le jeu est founi avec un guide touristique complet.

Modulae (Domaine Public)

Les programmeurs de Dijon ont encore frappé... Voici la suite de Nucleus, '*LA DEMO QUI TUE*'. Elle est encore plus spectaculaire que la précédente : animations en tout genre - animations vectorisées, en mode Fill, 3D... Venez la voir à Apple Expo. Les limites du GS ont encore été repoussées.

Addendum GS Epluché (ToolBox)

De trop nombreuses et regrettables erreurs étaient présentes dans ce livre de programmation consacré au IIGS. Une feuille d'errata est disponible auprès de votre éditeur préféré... Toutes les personnes ayant acheté le livre la recevront automatiquement à domicile.

ShrinKit GS (Domaine Public)

Le plus perfomant des compacteurs de fichiers et de disques existe maintenant en version full 16 bits. Le produit d'une qualité commerciale et pourtant diffusé en ShareWare est entièrement en mode desktop, et surtout, amélioration énorme, gère les fichiers étendus. Disponible chez tous les passionnés.

Design Master (Byte Works)

Design Master est un outil de programmation permettant de concevoir l'interface utilisateur d'un programme que vous développez grâce à des fonctions proches d'un logiciel de dessin ; Genesys (voir plus haut) est exactement du même esprit. Il est possible de créer des codes sources pour les langages suivants : Orca / Apw, Merlin, Orca / C, Orca / Pascal. Banc d'essai comparatif des deux logiciels dans un prochain numéro de ToolBox Mag. Pour le moment, nous donnerions plutôt l'avantage à Genesys...

Fumées sans feu

Jean-Yves Bourdin

«Ne croyez rien de ce que vous lisez, et pas plus du tiers de ce que vous voyez». Cette rubrique est conçue comme une illustration de ce proverbe. L'avertissement traditionnel "les propos tenus dans cette rubrique n'engagent que leur auteur etc." n'est pas suffisant.

Ces propos, imaginaires et sans fondement, n'engagent même pas leur auteur, ne serait-ce que parce que ledit auteur est incapable de résister au plaisir pervers de faire marcher Hubert Loiseleux –et quelques autres–, et de faire fulminer Bernard Fournier –et quelques autres– à propos de Sword of Sodan ou quelque autre poisson : on ne se refait pas…

S'il arrive que certaines des affirmations de cette rubrique se vérifient ensuite, ce serait pure coïncidence. Bref, ne croyez rien de ce que vous lisez dans cet article : dîtes-nous seulement si vous avez trouvé un intérêt à le lire.

Le Mac copie le GS

Les GSphiles qui n'ont jamais touché un Mac ne connaissent pas leur bonheur. Sur GS, qui veut rajouter un accessoire, une fonte, une init, un CDEV, un driver, un traducteur FST, ou même un outil, dans son système, se contente de copier le fichier concerné dans le sous-catalogue ad hoc : */SYSTEM/DESK.ACCS*, */SYSTEM/FONTS*, */SYSTEM/SYSTEM.SETUP*, */SYSTEM/CDEVS*, */SYSTEM/DRIVERS*, */SYSTEM/FST*, */SYSTEM/TOOLS* sont là pour ça. Pour choisir quels fichiers système seront actifs au boot, il suffit de les activer ou les désactiver avec le Finder, ProSel ou autre. Simple et propre.

Sur Mac en revanche, il faut introduire ou supprimer les fontes et les accessoires dans le *Fichier* System (pas le *Dossier*, le fichier : fontes et accessoires sont implantées en ressources de ce fichier). Il est courant qu'un fichier System fasse plusieurs mégas !

Or le Finder est incapable de faire ce travail. Sur GS, le Font/Da Mover du Maître Uhrich est un luxe bien confortable, et c'est un accessoire. Sur Mac, c'est une application séparée du Finder –et même pas au standard : elle n'a pas de menu Pomme et n'aime pas MultiFinder– qu'Apple met dans ses disques systèmes. Bien sûr, il y a des accessoires produits par les Uhrich du Mac. Mais enfin, le GS est, sur ce point, bien plus convivial.

Eh bien, il semble qu'Apple ait décidé, avec le système 7 pour Mac (à venir) de copier le GS : le Finder sera désormais capable de gérer fontes et accessoires, comme sur le GS. Plus besoin de Font/Da Mover.

Ceci dit, compatibilité oblige, ce sera toujours un fichier System, pas des dossiers. Les drivers, Inits temporaires, Inits permanentes, CDEVS, continueront à être jetés en vrac dans le dossier Système. Et ne demandez pas comment on fait pour rajouter un outil dans le fichier System du Mac !

Le GS copie le Mac

Une des nouveautés intéressantes que devrait introduire ce fameux système 7, mais sur les Macs les plus chers seulement, c'est la mémoire virtuelle : l'utilisation d'une partie du disque dur pour simuler de la mémoire vive.

L'idée a intéressé les concepteurs du futur système 6 du GS : le processeur du GS peut adresser 16 Mégas, mais le GS n'accepte que 8 Mégas réels en mémoire vive. Les autres 8 Mégas, c'est sur le disque dur qu'ils vont être.

Cela sera intégré à quelque chose que le Mac a eu avant nous : un MultiFinder, un programme qui permet d'avoir en mémoire plusieurs applications en même temps. L'idée était depuis longtemps bien avancée… en France (J.J. Leclerc).

Le Mac et le GS se copient

Apple a dans ses cartons une carte Mac Plus pour le GS. Il est donc quelque peu titillé par la carte Duet, de Cirtech, qui est un Mac // dans le GS. S'il la rachète à Cirtech, espérons que ça ne sera pas pour la mettre dans un placard...

Apple prépare aussi une carte Apple // pour ses nouveaux Macs. On parle pour l'instant d'une carte munie d'un 65C816, compatible GS donc, à mettre dans un des slots du mini-Mac II qui sortirait en Janvier.

Questions : sera-t-il possible de mettre une carte d'émulation Apple // sur la carte d'émulation Mac dans le GS ? Sera-t-il possible de mettre une carte d'émulation Mac sur la carte d'émulation Apple // dans le Mac ? Ce serait intéressant, un ordinateur qui s'émule lui-même par l'intermédiaire d'un autre...

En tout cas, avec HyperCard déjà compatible GS et Mac, la fusion pure et simple des deux gammes est aujourd'hui en cours. "Mac // GS", comme disait Sculley.

Et les petites annonces...

Les petites annonces sont entièrement gratuites pour les abonnés/membres de l'Association ToolBox. Pour les non-abonnés, les petites annonces sont facturées forfaitairement à un prix égal à celui de l'abonnement.

Votre annonce sera incluse dans la disquette du magazine. Du coup, vous avez le droit d'aller jusqu'à 50 lignes de 80 colonnes : le luxe...

Elle devra obligatoirement nous parvenir sous forme de fichier texte ASCII au standard GS/Mac ; fichier lisible directement par les logiciels GS tels AppleWorks-GS, BeagleWrite-GS, GraphicWriter III, etc. sans remplacement pour les caractères spéciaux ; sur disquette 800Ko ou sur cartouche Syquest SQ555 formatées en GS/OS. Les fichiers texte issus d'AppleWorks-8 ou dans un format autre que l'ASCII brut seront automatiquement rejetés, qu'on se le dise.

ToolBox Mag ne renverra pas la disquette ou la cartouche, et se garde le droit de ne pas passer votre annonce ; surtout si vous y incluez des considérations en désaccord avec la loi ou si vous polémiquez.

HyperCard GS

A la Conférence annuelle des Développeurs Apple //, qui s'est tenue au Kansas en Juillet, les participants ont signé un engagement de motus et bouche cousue total sur les produits Apple en cours de développement qui ont été présentés.

Ce qui suit étant une pure fiction, comme tout cet article, personne ne sera empalé si j'écris qu'HyperCard GS couleur ressemble de très près à la version MacIntosh –la nouvelle, la 2.0– et qu'un programme utilitaire appelé 'HyperMover' permet de transférer des piles entre les deux versions du programme.

C'est au point qu'on peut déjà entendre à propos d'HyperCard GS les griefs habituels contre HyperCard Mac : il serait difficile à utiliser, vorace en mémoire et en espace-disque, et lent.

Un peu de polémique

Tiens, cela permettra une petite mise au point sur ce genre de reproche qu'on entend encore trop souvent présentés comme griefs contre le GS.

D'abord, les logiciels d'aujourd'hui exigent plusieurs mégas de mémoire et un bon disque dur, et c'est normal. Au prix du méga de Ram de nos jours, avec la baisse des disques durs, il n'y a rien à dire contre cette évolution.

L'époque arrive où le fichier principal d'une application comme AppleWorks-GS, écrite en assembleur, ne tiendra plus sur une disquette 800Ko : l'Installer se chargera de copier les fragments du fichier depuis les disquettes 3,5" originales et de reconstituer le fichier entier sur le dur, comme c'est déjà le cas sur Mac avec PageMaker et autres. Ceux qui déplorent cette évolution veulent en fait qu'AppleWorks-GS ait moins de possibilités à offrir à l'utilisateur : ils veulent freiner le progrès.

Nous savons bien qui sont ceux qui mettent ainsi les deux pieds sur la pédale de frein : ce sont ceux qui sont bloqués sur les 512Ko de leur casserole ou les 640Ko de leur brouette. Qu'ils sachent seulement que nous qui avons choisi une machine ouverte, ne donnerions pas un clou de leur engin, et cela suffira sur ce point.

Quant à la lenteur, qu'on cesse de nous seriner que "X est plus rapide que le GS" ou qu'"HyperCard GS est plus lent qu'HyperCard Mac". Quelle que soit la machine –GS, Mac, ou... brouette– il y a deux choses qui sont tout simplement une question de gros sous : la vitesse de la machine, et sa résolution vidéo. Les GS lents sont lents, les Macs lents sont lents, les brouettes lentes... Les Macs rapides sont chers, les GS rapides sont chers, les "brouettes rapides", hum...

Bref, les Macs rapides et chers sont plus rapides et plus chers que les GS lents et bon marché, et les GS rapides et chers sont plus rapides et plus chers que les Macs lents et bon marché (mais si !). HyperCard GS est lent sur les GS lents et HyperCard Mac est lent sur les Macs lents. Si vous voulez que votre ordinateur, quel qu'il soit, soit plus rapide, ait une meilleure résolution, plus de couleurs, c'est simple : il suffit de payer.

En ce qui concerne HyperCard GS, le vrai débat est avec HyperStudio : le premier est le portage sur GS d'une application conçue pour le Mac Plus, la couleur y est un supplément, par exemple. Il n'est donc pas optimisé pour le

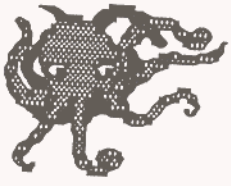
GS, mais permet de communiquer avec le Mac. Le second est optimisé pour le GS, mais ne sait rien faire des piles HyperCard. Mon choix dans ce genre de cas obéit à un principe simple : "les deux, mon adjudant !"

Donc, notre Emile national va avoir du travail : car HyperCard GS, nous le voulons.

Disques floptiques

Q/Cor, nouveau nom de Quadram, s'apprête à commercialiser un lecteur de disquettes magnéto-optiques 20 Mégas. Les disquettes, qui coûteraient dans les 25 dollars pièce, sont préformatées avec un "sillon optique" ineffaçable, qui guidera un laser pilotant les têtes de lecture-écriture magnétiques ; le problème avec nos floppys habituels est que la piste magnétique est toujours trop large.

Ce lecteur existera en version SCSI. Le temps d'accès moyen est de 35 ms, ce qui est dans la moyenne des temps d'accès des bons disques durs, et bien plus rapide que les CD Rom.



Zip GS

L'accélérateur Zip Chip GS est en bèta test. C'est une carte, pas une puce comme la Zip Chip //e, mais elle ne contient que 22 puces contre plus de 200 pour la Transwarp d'Æ. Elle tourne pour le moment à 12 Mhz.

Zip Technology sait désormais se taire aussi longtemps qu'il faut : il n'annoncera son produit que quand les problèmes de fabrication et de commercialisation seront réglé ; et nous savons que ce n'est pas son point fort. Mais rappelons que, finalement, la Zip Chip 8 bits a enterré tous les accélérateurs 8 bits concurrents, y compris la TransWarp III d'Æ, et reste désormais seule sur ce marché. Parfois, la patience est récompensée.

GS Plus ?

L'état actuel des prototypes du GS Plus doit vous inciter à faire des économies : car cela en vaudra la peine. La résolution vidéo de cet ordinateur serait portée à 600 sur 400, la vitesse serait de 5.8 Mhz, avec la possibilité d'une puce tournant bientôt à 7 Mhz.

Apple est en train d'expérimenter une nouvelle architecture dans la gestion des slots : il pourrait y avoir désormais des slots sur 16 bits ; la vitesse des slots pourrait se configurer par une option dans le tableau de bord. Il n'est pas exclu qu'une puce SWIM (Super Wozniak Integrated Machine) permette le branchement de lecteurs 3,5" haute-densité FDHD.

Economisez dès maintenant : il est probable qu'il vous faudra changer de moniteur couleur.

Claris, c'est Apple

Pourquoi donc Apple, qui possédait 80 % des parts de sa filiale Claris, a-t-il racheté les 20 % restants pour en faire une filiale à 100 % ? Les galériens, qui croient qu'il s'agissait d'empêcher Claris de programmer pour Windows sur IBM, se trompent, parce que Windows 3, eh bien, essayez-le donc : Blue is Blue, et on ne change pas par soft une brouette de chantier en Ferrari.

La raison est plus simple : qu'il s'agisse du GS ou du Mac, Claris est en fait complètement partie prenante dans les nouveaux systèmes d'Apple. Ce n'est pas par hasard que Claris sort une nouvelle version de ses logiciels en même temps qu'Apple sort un nouveau système. Claris a diffusé plus de GS/OS 5.02 avec AppleWorks-GS 1.1 qu'Apple n'en a vendus par ses concessionnaires.

Claris connait donc un bon paquet de "secrets d'Etat" du Mac et du GS. Et Apple ne veut pas que d'autres puissent acheter ces secrets pour fabriquer des compatibles, par exemple...

Bon, tout ça nous est assez indifférent. Mais ce qui suit l'est moins.

Il fut une époque où Apple France a très sincèrement essayé que Claris fasse une version française d'AppleWorks-GS. Ca n'a pas marché, parce que Claris, ce n'était pas Apple, et que Claris-France, ce n'était pas Claris, mais P-Ingénierie.

Bon, mais maintenant, Claris-France c'est Claris-France, et Claris c'est Apple. Ne serait-ce pas le moment de rappeler à Apple France qu'il fusille le GS en continuant imperturbablement à mettre dans la boîte l'inexcusable GS-Write ? AppleWorks-GS en français dans la boîte du GS, voilà le moyen de faire taire les critiques...

Des traîtres chez Beagle ?

Le premier "traître", c'est bien Beagle Bros lui-même, qui, après des années de fidélité à l'Apple //, a sorti l'an dernier "Flash", son premier logiciel Mac. Pas vraiment une réussite : Adobe vient en effet d'embaucher le programmeur de Flash, et c'était le seul programmeur Mac de Beagle.

On a bien commencé à en former d'autres, mais le temps de le faire, les fonctionnalités de Flash (un utilitaire de réseau) seront intégrées dans le système 7 du Mac...

OCR

Alan Bird et Rob Renstrom, ce n'est pas n'importe qui chez Beagle. Et pourtant, c'est avec Vitesse –les producteurs du Quickie– qu'ils se sont acoquinés pour fonder une nouvelle société qui sortira un logiciel de reconnaissance de caractères (compatible avec le Quickie de Vitesse et le LightningScan de ThunderWare immédiatement, mais aussi ultérieurement avec le scanner Apple, dont les drivers GS sont disponibles).

La société s'appelle WestCode Software, et le logiciel s'appelle InWords. Il sera capable de transformer des colonnes de texte imprimé en fichiers textes à la cadence de 3000 caractères par minute, ce qui est la vitesse d'une bonne dactylo. Une version tournera même sur //e !

Bref, Beagle sent bien que les TimeOut ne sont pas un filon illimité : AppleWorks, l'écran texte et le 8 bits, ce n'est guère l'avenir. Mais il cherche sa stratégie pour la suite... A notre avis, s'il regardait dans son fonds (l'héritage Beagle ancienne mode et celui de StyleWare), Beagle aurait largement de quoi produire les logiciels GS qu'on attend de lui.

Des Lasers pour le peuple

L'imprimante Laser se démocratise, les prix baissent. La pression est d'abord venue du côté d'Hewlett-Packard, avec ses Laser personnelles. Déjà la LaserJet III, de HP, munie du PostScript, est connectable directement sur le GS (munie d'un port Appletalk, elle utilise les drivers Apple).

Apple réplique avec ses nouvelles LaserWriter "personnelles", encore un peu chères pour qui veut PostScript.

PostScript, c'est ce langage de description de page d'Adobe, intégré à la LaserWriter, qui fait toute la beauté de Toolbox Mag et de bien d'autres. Mais voilà : il est cher.

On peut s'en tirer jusqu'ici, sur des imprimantes Laser non PostScript, par des émulations logicielles à mettre dans l'imprimante ou dans l'ordinateur, ou par des drivers d'imprimante sophistiqués.

Des drivers GS sont en préparation pour la HP LaserJet et d'autres imprimantes de qualité mais bon marché chez Seven Hills Software (l'éditeur de Graphic Writer III ; Bill Heineman, leur programmeur, est quelqu'un qui s'y connait...).

Adobe est déjà sérieusement concurrencé par ces émulateurs et drivers. Par ailleurs, il doit répliquer au nouveau standard de fontes non-Postcript d'Apple et de Microsoft (qui sera incorporé dans le système 7 du Mac). Il a donc décidé de fabriquer des contrôleurs PostScript bon marché, de façon à permettre de trouver sur le marché des imprimantes Laser PostScript à 1000 dollars.

Faîtes donc des économies : entre les GS super-accélérés, les Macs dans le GS et vice-versa, les scanners, les disquettes 20 Mégas, et les Laser abordables, le prochain Noël sera chaud...

■

Droits, reproduction, piratage etc.

Tout le contenu de la disquette et de la revue ToolBox Mag sont entièrement sous Copyright de l'Association ToolBox, et sont déposés à l'APP (Agence pour la Protection des Programmes).

Aucune reproduction, même partielle, et sous quelque forme que ce soit, de la disquette ou de la revue ne pourra avoir lieu sans accord préalable et écrit de ToolBox.

Les indications d'utilisation publiées dans la revue elle-même (voir pour InitTools et pour le Tool069 dans ce numéro) constituent éventuellement un tel accord.

Photocopier la revue et copier la disquette, c'est chercher à fusiller la seule revue française sur l'Apple IIGS : nous serions alors impitoyables...

Bidouilles GS

Jean-Yves Bourdin

The Graphic Exchange : patch GS/OS

La seule application qui soit sur mon disque dur et qui me force à rebooter en ProDOS 8 est "**The Graphic Exchange**", de Roger Wagner. Enervant.

Cette application est en effet écrite en code fixe, non relogeable par le Loader. Prévoyant, R. Wagner a mis sur la disquette plusieurs versions du programme, assemblées pour des bancs différents, et un petit programme lanceur qui regarde votre configuration mémoire et choisit la version à lancer. Ca marchait pour ProDOS 16, mais ça ne marchait plus pour le 5.0, qui bloque trop de bancs en bas de la mémoire.

Pourtant, TGE n'utilise que six bancs-mémoire en-dehors des bancs 00 et 01 : il devrait être possible d'en avoir une version qui "*monte*" plus haut, dans l'espace mémoire libre. L'idée était toute simple, mais il a fallu que j'attende qu'Yvan Kœnig me la donne : il suffit de comparer deux des versions différentes du programme pour avoir la liste des adresses et des valeurs de bancs.

Le mieux est de faire cela avec la commande "*EQUAL*" d'APW 1.1 : en profitant de la possibilité de rediriger sur un fichier disque les messages de la commande, vous obtenez un fichier texte dans lequel sont écrites sans fautes les valeurs et les adresses.

Vous récupérez ensuite ce fichier texte dans votre éditeur favori, et vous le transformez en un fichier texte qui devient un excellent programme AppleSoft. Ce programme AppleSoft vous permet de patcher TGE pour qu'il utilise les bancs mémoire que vous lui indiquerez, en fonction de votre configuration.

Pour constituer ce fichier, il suffit, sous APW 1.1, de taper :

```
EQUAL -D -N 1000 /TGE/EXCHANGE.P16 /TGE/EXCHANGE.ALT >&
/MON.DISQUE/MON.FICHIER
```

Note : il est possible que vous ayez à la fin du fichier un ensemble de différences sans importance, correspondant à des messages inutilisés. Supprimez-les de votre fichier texte, seules comptent les valeurs comprises entre $04 et $09.

Si vous avez la même version que moi de TGE (3.6), vous trouverez sur la disquette le fichier texte "*TGE.PATCHER*" qui marche sur ma version. Mettez-le sur le même disque qu'une copie du fichier "*EXCHANGE.P16*" de TGE, mettez sur le même disque le fichier appelé "*GRAPHIC.EXCHNGE*" de la disquette, passez sous Basic System, et tapez :

```
$ EXEC TGE.PATCHER
$ LIST 1000
```

Editez alors la ligne 1000, pour choisir la valeur de la variable A% : c'est elle qui détermine le premier des six bancs consécutifs que va utiliser TGE. Choisissez-la en fonction de votre configuration, à peu près aux deux tiers de votre espace-mémoire. Tapez "*RUN*", et vous aurez une version de TGE qui marche sous 5.0.

Note : ne lancez pas directement ce fichier, mais lancez plutôt le fichier "GRAPHIC.EXCHNGE" inclus sur la disquette. Voir ci-dessous pourquoi.

Sys.Launcher

Si vous lancez directement ce nouveau fichier "*EXCHANGE.P16*" par le Finder, celui-ci va vous dire "*Erreur 54*" et va vous envoyer dans les choux, ProSel-16, plus malin, refusera simplement de le lancer.

Cette erreur 54 est une erreur GS/OS, qui n'aime pas, cela fait partie de ses fantaisies, les fichiers ProDOS 8 trop longs. La solution a été trouvée par l'auteur de Shrinkit, qui a aussi un fichier P8 assez long : un petit fichier P8 lanceur qui lance le second.

Vous le trouverez sur la disquette sous le nom de "*GRAPHIC.EXCHNGE*". La finesse de ce petit fichier P8, c'est qu'il supporte le standard du Startup de Basic System. Si vous le lancez par le Finder, il faut éditer le chemin d'accès du fichier au début du programme, en $2007, avec octet de longueur en $2006. Sous ProSel 16, il suffit de mettre ce fichier, sous le nom de "*SYS.LAUNCHER*", dans le même catalogue que ProSel, et de mettre dans la rubrique "Startup"

des écrans ProSel le chemin d'accès complet du gros fichier ProDOS 8.

Votre écran de l'éditeur ProSel ressemblera à ceci :

```
                ProSel-16 Editor

-------------------------------------------------------
| Screen title :                                      |
|                                                     |
| Graphic Exchange                                    |
-------------------------------------------------------
| Prefix :                                            |
|                                                     |
|  ?                                                  |
-------------------------------------------------------
| Application path :                                  |
|                                                     |
| SYS.LAUNCHER                                        |
-------------------------------------------------------
| Startup :                                           |
|                                                     |
| /DUR/GRAPH/EXCHANGE.P16                             |
|                                                     |
-------------------------------------------------------

^D del rgt, DEL del lft, ^Y del to end, ^B begin, ^N end, ^R
replace,^O override cmds, --- save title, --- restore title.
ESC to exit. ^E toggles : Insert mode active.
```

Vous pourrez utiliser ce fichier pour lancer tous les programmes P8 qui engendrent une erreur 54 de GS/OS : il suffit de changer le Startup. Vous pouvez aussi remplacer "*SYS.LAUNCHER*" par "*PROSEL.SYSTEM*" de ProSel-8.

Bidouille pour les fontes

La chose vous étant expliquée par ailleurs dans ToolBox Mag, admettons-la simplement pour le moment : il y a plusieurs sortes de fontes différentes sur le GS. En tout cas, vous devez avoir des stocks différents pour les fontes destinées à l'affichage sur écran 320, pour celles destinées à l'affichage sur écran 640, pour celles destinées à l'impression sur Imagewriter, et pour celles destinées à représenter sur écran les fontes de la Laser.

Bien sûr, vous pouvez tout mettre, en vrac, dans */SYSTEM/FONTS*. Mais alors, quel bazar au moment de choisir sa fonte. Il faut se rappeler que Shaston 16 est destiné à l'affichage écran 640, Courier à la Laser, etc. Bref, le menu des fontes descend jusqu'en bas, et est encombré de fontes inutiles.

La solution serait évidemment d'avoir plusieurs catalogues de fontes différents, et pas seulement */SYSTEM/FONTS*. Bien sûr, le Font/DA Mover du Maître Uhrich permet de mettre le minimum dans */SYSTEM/FONTS*, et de charger et évacuer les autres fontes en fonction des besoins. Mais j'aime bien les solutions toutes faites.

La mienne consiste à avoir des sous-catalogues */SYSTEM/FONTA*, */SYSTEM/FONTB*, etc, en fonction des besoins et des applications. Comment faire pour que le GS les reconnaisse ? Eh bien, je patche "à la volée" le Font Manager (Tool27), par une seule ligne de programme Basic (le fait d'être sous AppleSoft et P8 permet d'être certain que le Font Manager n'est plus en Ram au moment où on le patche).

Voici cette ligne pour faire reconnaître */SYSTEM/FONTA* :

```
$ 100 POKE 768, 65 : PRINT CHR$(4)
;"BSAVE/DUR/SYSTEM/TOOLS/TOOL027, T$BA, A$300, B$19E"
```

Modifiez le Poke du début pour remettre */SYSTEM/FONTS* ou passer à */SYSTEM/FONTB*. Le "*B$19E*" correspond à l'adresse concernée sur le Font Manager du système 5.02 français. Elle est différente pour d'autres systèmes. Par pitié, ne demandez plus qu'Apple mette les Tools en Rom, parce qu'alors, finis les patches !

Dessins digitalisés par Quickie et imprimés sur LaserWriter II à partir d'AppleWorks GS

Courrier des Lecteurs

Jean-Yves Bourdin

Evidemment, le Courrier des lecteurs du numéro 1 est un peu maigre. Profitons-en pour définir quelques procédures.

① Nous essaierons de répondre à tout courrier dans un délai... pas trop long. Si votre courrier contient une enveloppe timbrée à votre adresse, vous aurez une réponse personnelle.

Sinon, réponse dans le magazine, rappel : "magazine", cela peut vouloir dire revue ou disquette. Si vous êtes pressé, téléphonez : ToolBox assure un support téléphonique des programmes et articles de ToolBox Mag.

② Si votre courrier est adressé à un des auteurs de ToolBox Mag, mettez clairement son nom sur l'enveloppe. Nous ferons suivre.

③ Si vous avez un problème de type technique, pensez à mentionner clairement quelle est votre configuration, matérielle et logicielle.

Sachez que ToolBox Mag n'assure le support que de deux systèmes d'exploitation pour l'Apple IIGS : ProDOS 8 et GS/OS dans sa version française la plus récente –GS/OS 5.02 au moment où nous écrivons ces lignes– et que de deux types d'Apple IIGS : l'Apple IIGS Rom 01 avec 1,2 Mégas de mémoire minimum, et l'Apple // GS Rom 03.

④ Pour nos auteurs, quelques procédures de transmission

✏ Gardez un double de **tout** ce que vous nous envoyez : on ne sait jamais...

✏ Etiquetez toute disquette avec votre nom, votre adresse et la date d'envoi et/ou la version de votre programme.

✏ Pour les textes, vous pouvez mettre des fichiers formatés ; nous préférons les fichiers traitement de texte –et non PAO– d'AppleWorks-GS. Mais mettez obligatoirement une version en texte pur, fichiers Ascii sans Return en fin de ligne, tels ceux qu'exporte AppleWorks-GS. C'est nous qui ferons la mise en pages.

✏ Envoyez-nous **tout** ce qui est nécessaire pour que nous puissions réassembler ou recompiler vos programmes. N'oubliez pas que vous avez sans doute bidouillé à votre façon vos Equates, vos Librairies, etc.

✏ Sachez à l'avance que vous serez surpris par ce qui sera publié : notre nouvelle formule de revue "disquette incluse" nous permet de répartir à notre façon votre envoi entre papier et disquette. De plus, étant nouvelle, cette formule n'est pas du tout rodée...

✏ Si vous êtes publié, vous toucherez vos droits d'auteur dans la quinzaine qui suit la parution de ToolBox Mag. Si vous n'êtes pas abonné, nous commencerons par en déduire le montant de votre abonnement ; ToolBox Mag est un vrai magazine avec un suivi rédactionnel : vous ne pourrez y insérer valablement votre contribution que si vous le lisez.

Notez que la rémunération prévue pendant la période de lancement de ToolBox Mag est franchement insuffisante. Mais le succès de ToolBox Mag nous permettra progressivement de l'augmenter, et vous allez bientôt vous enrichir, n'en doutez pas.

✏ Si vous n'êtes pas publié aussitôt, ne désespérez pas : il est fort possible que nous ayons réservé votre contribution pour un numéro ultérieur, un numéro spécial, une édition séparée, etc. Contactez-nous éventuellement.

⑤ Donnez votre avis

D'ores et déjà, lecteurs, nous souhaitons avoir votre avis le plus vite possible sur l'orientation de la revue :

✏ Souhaitez-vous que nous consacrions une place substantielle aux jeux ? Plutôt des revues logicielles, plutôt des trucs pour jouer, plutôt des patches pour tricher ?

✏ Devons-nous accorder une place au 8 bits (AppleWorks 8 et UltraMacros), et si oui, laquelle ? Souhaitez-vous des programmes en AppleSoft ?

✏ Quelle place pour l'HyperMédia (HyperStudio etc), et sous quelle forme (initiation, Xcmds, etc) ?

■

```
MJBootInit START
        using _MJukeDt    ; initialisation de l'outil
        lda   #0
        sta   >MJOpen     ; outil fermé
        sta   >MJErHn     ; gestion normale des erreurs
        sta   >NextAdr    ; pas de morceaux chargés
        sta   >NextAdr+2
        clc
        rtl
        END
MJStatus START
        using _MJukeDt    ; demande de l'état de MJUKE
        lda   >MJOpen     ; lecture de l'état
        sta   7,s         ; renvoi du paramètre
        lda   #0          ; pas d'erreur
        clc
        rtl
        END
MJVersion START
                          ; demande de version
        lda   #$0101      ; renvoie la version 1.01
        sta   7,s
        lda   #0
        clc
        rtl
        END
MJReset  START
        using _MJukeDt   ; RAZ de MJUKE
        lda   #0
        sta   >MJErHn    ; RAZ de la gestion d'erreur
        sta   >NextAdr   ;  et du morceau courant
        sta   >NextAdr+2
        jsl   MJStop     ; arrêt du morceau courant
        clc
        rtl
        END
_MJdeath PRIVATE
                          ; erreur fatale
        pha
        pea   $0000       ; message standard
        pea   $0000
        ldx   #$1503
        jsl   $E10000     ; pas de retour après
        END               ; cet appel
_MJAlert PRIVATE
                          ; affiche la fenêtre
        using _MJukeDt    ; d'alerte d'erreur disque
        ldx   #disker     ; adresse de la ligne
        stx   _MJmess     ; à afficher
        ldx   #^disker
        stx   _MJmess+2
_MJAlerr ENTRY
                          ; affiche une fenêtre
        ldx   MJErHn      ; d'alerte générale
        bne   aal
        sec               ; retour si gestion non
        rts               ; automatique des erreurs
                          ; voir MJStartUp
aal     pha
        PushWord #0
        PushWord #$FFFF
        PushWord #0
        _FlushEvents      ; vide la file d'événements
        pla
        _ResetAlertStage  ; RAZ du niveau d'alerte
        pla               ; conversion en hexa du
        PushLong #0       ; code d'erreur
        pha
        _HexIt
        _MJerror
        pla
        sta   code+2
        pla
        sta   code+4
        PushWord #0       ; affiche la fenêtre
        PushLong #error
        PushLong #0
        _CautionAlert
        _MJerror          ; pas d'erreur ?
        pla
        lda   #0
        clc
        rts
disker  str   "Erreur Prodos !"
error   anop
        dc    i2'50,172,100,468,1' ; fenêtre d'alerte
        dc    i2'$81,$81,$81,$81'  ; un seul bip
        dc    i4'butok,_MJerr,coderr,0'
coderr  anop              ; code d'erreur
        dc    i2'3,20,70,30,290,$800F'
        dc    i4'code,0,0'
code    dc    i1'5'       ; ligne à afficher
        dc    c"$    "
butok   anop              ; bouton "OK"
        dc    i2'1,32,110,46,190,$0A'
        dc    i4'oktext'
        dc    i2'0,1'
        dc    i4'0'
oktext  str   "OK"
        END
_MJukeDt  PRIVDATA
                          ; données diverses
_MJerr  dc    i2'2,10,70,20,290,$800F'
_MJmess dc    i4'0,0,0'
_MJID   ds    2           ; Memory ID
_MJDP   ds    2           ; adresse de la page zéro
_MJsong dc    i2'0'       ; est-ce qu'une musique est jouée
_MJwavef ds   16          ; nom du WaveBank
MJOpen  dc    i2'0'       ; est-ce que MJUKE est ouvert ?
MJErHn  ds    2           ; type de gestion d'erreur
NextSong ENTRY
        dc    i1'$5C'
NextAdr ds    4           ; adresse de la routine de
gestion
                          ;  du Juke-box
                          ; messages d'erreurs
JukeErr str   'Fichier Juke-box incorrect !'   ; $4502
NotWErr dc    i1'29'                           ; $4503
        dc    c'Fichier Wavebank non charg'
        dc    i1'$8E'
        dc    c' !'
notfound str  'Fichier wavebank introuvable !' ; $4504
WVerr   str   'fichier Wavebank incorrect !'   ; $4505
MountErr str  'Pas plus de 40 musiques !'      ; $4506
PrefErr dc    i1'29'                           ; $4507
        dc    c'Num'
        dc    i1'$8E'
        dc    c'ro de pr'
        dc    i1'$8E'
        dc    c'fixe incorrect !'
        END
                          ; variables en page zéro
SeqPtr  gequ  $0          ; adr de la prochaine note
rptptr  gequ  $4
delta   gequ  $6          ; temps avant prochaine note
ntaill  gequ  $8
note    gequ  $A
inst    gequ  $C
duree   gequ  $E
GenCount gequ $10
GenNote gequ  $30
insttable gequ $4C
volume  gequ  $50
repet   gequ  $54
gen     gequ  $64
osctable gequ $66
song    gequ  $6A         ; musique courante
box     gequ  $6C         ; jouer plusieurs morceaux ?
songaval gequ $6E         ; nb de musiques chargées
song2   gequ  $70
timercnt gequ $72         ; timer en 100èmes de secondes
timercn2 gequ $74
namebuf gequ  $76
        gequ  $7A
MJJuke  START
        using _MJukeDt    ; lancement du Juke-box
        lda   >_MJDP      ; fixe la page zéro
        phd
        tcd
        lda   #1
        sta   box         ; Juke-box = on
        pld
        bra   _MJPLAY1
MJPlay  ENTRY
        lda   >_MJDP
        phd
        tcd
        stz   box         ; Juke-box = off
        pld
```

```
_MJPLAY1 ENTRY
         lda   7,s          ; lecture des paramètres
         tay
         lda   5,s
         sta   7,s
         lda   3,s
         sta   5,s
         lda   1,s
         sta   3,s
         pla
_MJPlay2 ENTRY
         phb
         phk
         plb
         phd
         lda   _MJDP
         tcd
         sty   song2
         lda   songaval
         bne   p1
         pld                ; pas de morceau chargé
         plb
         lda   #0
         clc
         rtl
p1       anop
         stz   delta        ; arrêt du séquenceur
         _CouperNotes       ; arrêt des notes
         ldx   #$62         ; RAZ des variables
p2       stz   0,x
         dex
         dex
         bpl   p2
         lda   song2
         inc   a
         beq   p3           ; même morceau
         dec   a
         beq   p4           ; morceau suivant
         dec   a
         sta   song
         bra   p3
p4       anop
         lda   NextAdr
         ora   NextAdr+2
         beq   p401
         phd
         phb
         pea   $0000
         lda   song
         inc   a
         pha
         jsl   NextSong     ; routine utilisateur
         long
         pla
         plb
         pld
         inc   a
         beq   p3           ; même morceau
         dec   a
         beq   p401         ; morceau suivant
         dec   a
         sta   song
         bra   p3
p401     inc   song
p3       lda   song
         cmp   songaval
         bcc   p5
         stz   song         ; retour au 1er morceau
p5       anop
         lda   song         ; calcul de l'adr du
         asl   a            ; handle du morceau
         adc   song
         tax
         lda   $80,x        ; adr du handle
         sta   SeqPtr
         lda   $82,x
         and   #$00FF
         sta   SeqPtr+2
         lda   [SeqPtr]     ; adr du morceau
         tax
         ldy   #2
         lda   [SeqPtr],y
         sta   SeqPtr+2
         stx   SeqPtr
         ldy   #54
         lda   [SeqPtr],y   ; volume sonore
         and   #$00FF
         sta   volume
         ldy   #24
         ldx   #0
p6       lda   [SeqPtr],y   ; offsets des instruments
         clc                ; calcul des adr des
         adc   SeqPtr       ; instruments
         sta   instadr,x
         lda   SeqPtr+2
         adc   #0
         sta   instadr+2,x
         iny
         iny
         inx
         inx
         inx
         inx
         cpy   #54
         bne   p6
         lda   #instadr
         sta   insttable
         lda   #^instadr
         sta   insttable+2
         lda   SeqPtr
         clc
         adc   #55
         sta   SeqPtr       ; adr des notes
         lda   SeqPtr+2
         adc   #0
         sta   SeqPtr+2
         stz   timercnt     ; RAZ du timer
         stz   timercn2
         lda   #1
         sta   _MJsong      ; on joue un morceau
         sta   delta        ; lance le morceau
         pld
         plb
         lda   #0
         clc
         rtl
instadr  ds    15*4
         END
MJNext   START
                            ; installe la routine
         using _MJukeDt     ; d'enchainement des musiques
         lda   7,s
         sta   >NextAdr
         lda   9,s
         sta   >NextAdr+2
         lda   5,s
         sta   9,s
         lda   3,s
         sta   7,s
         lda   1,s
         sta   5,s
         pla
         pla
         lda   #0
         clc
         rtl
         END
MJMount  START
         using _MJukeDt     ; charge un morceau
         lda   7,s          ; lit les paramètres
         sta   >prefixe
         lda   9,s
         sta   >picdest
         lda   $B,s
         sta   >picdest+2
         lda   5,s
         sta   $B,s
         lda   3,s
         sta   9,s
         lda   1,s
         sta   7,s
         pla
         pla
         pla
         phb
         phk
         plb
         lda   prefixe
         cmp   #8
         bcc   e0
         lda   #PrefErr     ; préfixe incorrect
         sta   _MJmess
         lda   #^PrefErr
         sta   _MJmess+2
         lda   #$4507
         jsr   _MJAlerr
```

```
        plb
        rtl
e0      lda   _MJDP
        phd
        tcd
        lda   picdest
        sta   namebuf
        lda   picdest+2
        sta   namebuf+2
        lda   [namebuf]
        and   #$00FF
        tay
        dey
        lda   MCfile+2    ; place le suffixe ".MJK"
        sta   [namebuf],y
        dey
        dey
        lda   MCfile
        sta   [namebuf],y
        lda   songaval
        pld
        cmp   #40
        bcc   e1
MountEr lda   #MountErr  ; pas plus de 40 morceaux !
        sta   _MJmess
        lda   #^MountErr
        sta   _MJmess+2
        lda   #$4506
        jsr   _MJAlerr
        plb
        rtl
e1      anop
        _OPEN Params      ; ouvre le fichier
        bcc   e2
        jsr   _MJAlert
        plb
        rtl
e2      _GET_EOF Params   ; lit sa longueur
        bcc   e3
        pha
        _CLOSE Params
        pla
        jsr   _MJAlert
        plb
        rtl
e3      lda   picdest
        sta   taill
        lda   picdest+2
        sta   taill+2
        PushLong #0
        PushLong taill
        PushWord _MJID
        PushWord #$C000
        PushLong #0
        _NewHandle        ; réserve la mémoire
        _MJerror
        pla
        sta   PHndl
        pla
        sta   PHndl+2
        tsc
        sec
        sbc   #8
        tcs
        phd
        inc   a
        tcd
        ldy   #2
        lda   [4],y
        sta   picdest+2
        tax
        lda   [4]
        sta   picdest
        sta   4
        stx   6
        _READ Params      ; lit le fichier
        bcc   e4
        _CLOSE Params
        jsr   _MJAlert
        brl   abort2
e4      CLOSE Params      ; et enfin le ferme
        lda   [4]
        cmp   juke
        bne   JukeEr
        ldy   #2
        lda   [4],y
        cmp   juke+2
        beq   e5
JukeEr  lda   #JukeErr    ; format incorrect !
        sta   _MJmess
        lda   #^JukeErr
        sta   _MJmess+2
        lda   #$4502
        jsr   _MJAlerr
        brl   abort2
e5      lda   4
        clc
        adc   #8
        sta   0
        lda   6
        adc   #0
        sta   2
        lda   _MJDP
        phd
        tcd
        lda   songaval
        asl   a
        adc   songaval
        tax
        lda   PHndl
        sta   $80,x
        lda   PHndl+2
        shortm
        sta   $82,x
        longm
        pld
        txa
        beq   e8
        shortm            ; teste si le Wavebank
        lda   [0]         ; est déjà chargé
        cmp   _MJwavef
        bne   NotWave
        tay
e7      lda   [0],y
        cmp   _MJwavef,y
        bne   NotWave
        dey
        bne   e7
        longm
        pld
        tsc
        clc
        adc   #8
        tcs
        brl   e16
NotWave longm             ; Wavebank non chargé
        lda   #NotWErr
        sta   _MJmess
        lda   #^NotWErr
        sta   _MJmess+2
        lda   #$4503
        jsr   _MJAlerr
        brl   abort2
e8      anop
        shortm
        lda   [0]
        cmp   _MJwavef
        bne   e81
        tay
e70     lda   [0],y
        cmp   _MJwavef,y
        bne   e81
        dey
        bne   e70
        longm
        pld
        tsc
        clc
        adc   #8
        tcs
        brl   e160
e81     long
        ldy   #$E
e9      lda   [0],y
        sta   _MJwavef,y
        dey
        dey
        bpl   e9
        lda   [0]
        tax
        lda   4
        clc
        adc   #6
        sta   0
        lda   6
        adc   #0
```

```
        sta   2
        ldy   #1
        lda   prefixe
        ora   #$2F30      ; transforme le prefixe en digit
        sta   [0],y
        txa
        shortm
        inc   a
        inc   a
        sta   [0]
        longm
        lda   0
        sta   picdest
        lda   2
        sta   picdest+2
        _OPEN Params      ; ouvre le Wavebank
        bcc   e11
        ldx   #0
        stx   _MJwavef
        cmp   #$46        ; 'file not found'
        beq   e10
        jsr   _MJAlert
        brl   abort2
e10     lda   #notfound   ; Wavebank introuvable !
        sta   _MJmess
        lda   #^notfound
        sta   _MJmess+2
        lda   #$4504
        jsr   _MJAlerr
        brl   abort2
e11     _GET_EOF Params
        bcc   e12
        pha
        ldx   #0
        stx   _MJwavef
        _CLOSE Params
        pla
        jsr   _MJAlert
        brl   abort2
e12     lda   picdest
        sta   taill
        lda   picdest+2
        sta   taill+2
        PushLong #0
        PushLong taill
        PushWord _MJID
        PushWord #$C000
        PushLong #0
        _NewHandle
        _MJerror
        pla
        sta   PHndl
        sta   0
        pla
        sta   PHndl+2
        sta   2
        ldy   #2
        lda   [0],y
        sta   picdest+2
        tax
        lda   [0]
        sta   picdest
        sta   0
        stx   2
        _READ Params      ; charge le Wavebank
        bcc   e13
        pha
        ldx   #0
        stx   _MJwavef
        _CLOSE Params
        pla
        jsr   _MJAlert
        brl   abort2
e13     _CLOSE Params
        lda   [0]
        cmp   WVOk
        bne   WVer
        ldy   #2
        lda   [0],y
        cmp   WVOk+2
        beq   e14
WVer    ldx   #0          ; Wavebank incorrect !
        stx   _MJwavef
        lda   #WVerr
        sta   _MJmess
        lda   #^WVerr
        sta   _MJmess+2
        lda   #$4505
        jsr   _MJAlerr
        brl   abort2
e14     ldy   #4
        lda   [0],y
        clc
        adc   0
        tax
        lda   2
        adc   #0
        php
        pha
        phx
        PushWord #0
        PushWord #$0
        sei
        _WriteRamBlock    ; transfère le Wavebank
        _MJerror          ;  en mémoire son
        plp
e15     pld
        tsc
        clc
        adc   #8
        tcs
        PushLong PHndl
        _DisposeHandle    ; libère la mémoire vive
        _MJerror
e160    PushWord #250
        PushLong #_MJSeq
        _NSStartUp        ; ouvre le Note Synthesizer
        _MJerror
e16     lda   _MJDP
        phd
        tcd
        inc   songaval    ; un morceau de plus
        pld
        plb
        lda   #0
        clc
        rtl
abort2  anop              ; abandon du chargement
        pha
        PushLong PHndl
        _DisposeHandle
        plx
        pld
        tsc
        clc
        adc   #8
        tcs
        plb
        txa
        cmp   #1
        rtl
Params  anop
readid  ds    2
picdest ds    4
taill   ds    4
realc   ds    4

WVOk    dc    c'GSWV'
PHndl   ds    4
prefixe ds    2
MCfile  dc    c'.'
juke    dc    c'MJK2'
        END
MJStop  START
        using _MJukeDt    ; arrêt du séquenceur
        lda   >_MJsong
        beq   s0
        lda   >_MJDP
        phd
        tcd
        stz   delta       ; arrêt des notes
        _CouperNotes      ; et du son
        pld
s0      lda   #0
        clc
        rtl
        END

MJShutDown  START
        using _MJukeDt    ; ferme MJUKE
        lda   >MJOpen
        beq   n2
        lda   >_MJDP
        phd
        tcd
        stz   delta       ; coupe le son
        _CouperNotes
```

```
        PushWord #0
        _NSStatus          ; ferme le Note Synthesizer
        _MJerror
        pla
        beq   q1
        _NSShutDown
        _MJerror
q1      anop
        _SoundShutDown     ; ferme le Sound Manager
        _MJerror
n1      lda   songaval
        beq   n3
        dec   songaval
        dec   a
        asl   a
        adc   songaval
        tax
        lda   $82,x
        and   #$00FF
        pha
        lda   $80,x
        pha
        _DisposeHandle     ; libère toute la mémoire
        _MJerror
        bra   n1
n3      pld
n2      lda   #0
        sta   >MJOpen
        clc
        rtl
        END
MJCont  START
                           ; continue le morceau
        using _MJukeDt     ; interrompu
        lda   >_MJsong
        beq   c1
        phd
        lda   >_MJDP
        tcd
        lda   #1
        sta   delta        ; relance le séquenceur
        pld
c1      lda   #0
        clc
        rtl
        END
MJStartUp  START
        using _MJukeDt     ; initialise MJUKE
        lda   7,s
        sta   >MJErHn
        lda   9,s
        sta   >_MJID
        lda   5,s
        sta   9,s
        lda   3,s
        sta   7,s
        lda   1,s
        sta   5,s
        pla
        pla
        lda   >MJOpen
        beq   a10
        lda   #$4501       ; MJUKE déjà initialisé !
        sec
        rt
a10     phb
        phk
        plb
        PushWord #25
        PushWord #$100
        _LoadOneTool       ; charge le TOOL025
        _MJerror           ; (Note Synthesizer)
        stz   _MJsong
        PushLong #0
        PushLong #$200
        PushWord _MJID
        PushWord #$C001
        PushLong #0
        _NewHandle         ; réserve deux pages zéro
        _MJerror
        pla
        pla
        tsc
        sec
        sbc   #4
        tcs
        phd
        inc   a
        tcd
        lda   [0]
        tax
        clc
        adc   #$100
        sta   _MJDP
        pld
        tsc
        clc
        adc   #4
        tcs
        phx
        _SoundStartup      ; ouvre Sound Manager
        _MJerror
        lda   _MJDP        ; RAZ de la 2ème page zéro
        phd
        tcd
        ldx   #$FE
a11     stz   0,x
        dex
        dex
        bpl   a11
        ldx   #$E
a12     stz   _MJwavef,x   ; RAZ du nom du Wavebank
        dex
        dex
        bpl   a12
        PushLong #0
        _GetTableAddress   ; lit l'adr de la table
        _MJerror           ; d'accès aux oscillateurs
        PullLong osctable
        ldy   #4
        lda   [osctable],y
        sta   _MJadr
        iny
        iny
        shortm
        lda   [osctable],y
        sta   _MJadr+2
        longm
        ldy   #$18
        lda   [osctable],y
        tax
        iny
        iny
        lda   [osctable],y
        sta   osctable+2
        stx   osctable
        pld
        lda   #$FFFF
        sta   MJOpen       ; MJUKE = ouvert
        plb
        lda   #0
        clc
        rtl
_MJreg  ENTRY
        dc    i1'$5C'      ; JMP absolu long
_MJadr  ds    3            ; adresse du JMP long ci-dessus
        END
MJWave  START
        using _MJukeDt     ; changement de Wavebank
        jsl   MJStop       ; arrêt du morceau courant
        phb
        phk
        plb
        phd
        lda   _MJDP
        tcd
        stz   box
n1      lda   songaval
        beq   newend
        dec   songaval
        dec   a
        asl   a
        adc   songaval
        tax
        lda   $82,x
        and   #$00FF
        pha
        lda   $80,x
        pha
        _DisposeHandle     ; libère toute la mémoire
        _MJerror
        bra   n1
newend  PushWord #0
        _NSStatus          ; ferme le Note synthesizer
```

```
        _MJerror
        pla
        beq   n2
        _NSShutDown
        _MJerror
n2      anop
        pld
        plb
        lda   #0
        clc
        rtl
        END
MJTimer START
                          ; lit le timer
        using _MJukeDt    ;  ( en 1/10s )
        lda   >_MJDP
        phd
        tcd
        lda   timercnt
        pld
        sta   7,s
        lda   #0
        clc
        rtl
        END
MJSong  START
        using _MJukeDt    ; lit le morceau courant
        lda   >_MJDP
        phd
        tcd
        lda   song
        inc   a
        pld
        sta   7,s
        lda   #0
        clc
        rtl
        END
                          ;
                          ;
                          ; Sequenceur 1 piste
                          ;
                          ;
_MJSeq  PRIVATE
        using _MJukeDt
        long              ; fixe la page zéro
        phd
        lda   >_MJDP
        tcd
        lda   delta
        bne   t1
        brl   nonotes     ; séquenceur arrêté
t1      inc   timercn2    ; gestion du timer
        ldx   timercn2
        cpx   #10
        bcc   t12
        inc   timercnt
        stz   timercn2
t12     dec   a
        sta   delta
        beq   event
        brl   nonotes
event   anop              ; des notes à jouer
        stz   ntaill
        stz   note
        stz   inst
        lda   [SeqPtr]    ; lit la durée avant
        and   #$00FF      ; la prochaine note
        cmp   #$FE
        bcc   t3
        beq   t2
        brl   eventlong   ; note longue
t2      brl   repeat      ; répétition
t3      sta   delta       ; note courte
        short
        ldy   #1
        lda   [SeqPtr],y  ; lit le demi-ton
        sta   note
        iny
        lda   [SeqPtr],y
        sta   inst        ; lit l'instrument
        iny
        lda   [SeqPtr],y  ; lit la durée
        long
        and   #$00FF
        sta   duree
        bra   eventtype
endsong stz   delta       ; fin du morceau
        lda   #$FFFF
        sta   timercnt    ; timer = $FFFF ou -1
        lda   box         ; si Jukebox = ON, alors
        bne   next        ; on joue le morceau suivant
        pld
        rtl
next    pld
        ldy   #0
        jsl   _MJPlay2    ; morceau suivant
        rtl
eventlong anop            ; note longue (7 octets)
        inc   ntaill
        ldy   #$1
        lda   [SeqPtr],y  ; durée avant la note suivante
        cmp   #$FFFF
        beq   endsong
        sta   delta
        short
        iny
        iny
        lda   [SeqPtr],y  ; lit le demi-ton
        sta   note
        iny
        lda   [SeqPtr],y  ; lit l'instrument
        sta   inst
        iny
        long
        lda   [SeqPtr],y  ; lit la durée
        sta   duree
        cmp   #$FFFF
        bne   eventtype   ; note infinie
        lda   inst
        and   #$007F
        ora   #$FF00
        sta   duree
eventtype shortm
        lda   inst
        beq   nosound     ; silence
        bpl   t4
        brl   endnote
t4      longm
        PushWord #0       ; place pour le résultat
        PushWord #64      ; priorité
        _AllocGen         ; alloue le générateur
        lda   1,s
        tay
        asl   a
        sta   gen
        lda   [osctable],y
        and   #$00FF
        clc
        adc   #$A0
        tax
        shortm
        lda   #$03
        jsl   _MJreg      ; coupe le générateur
        inx               ;  au cas où il fonctionne
        jsl   _MJreg
        longm
        ldx   gen
        lda   duree
        sta   GenCount,x
        lda   note
        tay
        and   #$007F
        pha
        sta   GenNote,x
        lda   volume
        cpy   #$0080
        bcc   t5
        lda   #$007F
t5      pha
        lda   inst
        and   #$007F
        dec   a
        asl   a
        asl   a
        tay
        lda   [insttable],y
        tax
        iny
        iny
        lda   [insttable],y
        pha
        phx
        _NoteOn           ; joue la note
nosound anop              ; calcule l'adr de
```

```
        longm              ; la note suivante
        lda   ntaill
        beq   court
tlong   lda   SeqPtr       ; note longue
        clc
        adc   #7
        sta   SeqPtr
        bcc   t11
        inc   SeqPtr+2
t11     lda   delta
        bne   exit
        brl   event
court   anop               ; note courte
        lda   SeqPtr
        clc
        adc   #4
        sta   SeqPtr
        bcc   t10
        inc   SeqPtr+2
t10     lda   delta
        bne   exit
        brl   event
exit    anop
        bra   nonotes
endnote longm              ; fin de note infinie
        lda   inst
        and   #$007F
        ora   #$FF00
        sta   inst
        ldy   note
        ldx   #15*2
t6      cmp   GenCount,x   ; cherche le bon générateur
        bne   t8
        pha
        tya
        cmp   GenNote,x
        bne   t7
        lda   duree
        sta   GenCount,x   ; nouvelle durée
        pla
        bra   nosound
t7      pla
t8      dex
        dex
        bpl   t6
        bra   nosound
nonotes ldx   #15*2        ; gestion de la fin des notes
genloop lda   GenCount,x
        beq   notdone
        tay
        and   #$FF00
        cmp   #$FF00
        beq   notdone
        tya
        dec   a
        sta   GenCount,x
        bne   notdone
        phx
        txa
        lsr   a
        pha
        lda   GenNote,x
        pha
        _NoteOff           ; fin de note
        plx
notdone anop
        dex
        dex
        bpl   genloop
        pld
        rtl
repeat  anop               ; gestion des répétitions
        ldy   #1
        short
        lda   [SeqPtr],y
        tax
        iny
        lda   [SeqPtr],y
        cmp   repet,x
        bcc   rptfini
        inc   repet,x
        long
        iny
        lda   SeqPtr
        sec
        sbc   [SeqPtr],y
        sta   SeqPtr
        lda   SeqPtr+2
        sbc   #0
        sta   SeqPtr+2
        brl   event
rptfini stz   repet,x      ; fin de répétition
        long
        lda   SeqPtr
        clc
        adc   #5
        sta   SeqPtr
        bcc   t9
        inc   SeqPtr+2
t9      brl   event
        END
-----------------------------------------------------------
* Macros pour MJuke
        MACRO
&lab    _MJerror
&lab    bcc   END&SYSCNT
        brl   _MJdeath
END&SYSCNT anop
        MEND
        MACRO
&lab    _CouperNotes
&lab    ldx   #$0D19
        jsl   $E10000
        MEND
        MACRO
&lab    errordeath
&lab    bcc     END&SYSCNT
        brl     sysdeath
END&SYSCNT anop
        MEND
-----------------------------------------------------------
```

Exemple de programme en TML Pascal utilisant MJuke

```
PROGRAM  MJExempl;
        { Exemple sous TML PASCAL 1.00D en mode texte }
USES    QDIntf, GSIntf, MJuke;
var     i : integer;
{ Routine de gestion de l'enchaînement des morceaux
  ils sont joués dans l'ordre 1 - 3 - 2 - 4 - ...
  Cette routine ne sert qu'à titre d'exemple et peut être
  supprimée sans problème, à condition de supprimer aussi
l'ordre MJNext}
{$LongGlobals+}
Function {$DefProc} RePlay(morceau: integer): integer;
var     j : integer;
begin
        if morceau = 1 then j := 3;
        if morceau = 2 then j := 4;
        if morceau = 3 then j := 2;
        if morceau > 3 then j := morceau + 1;{ on joue le
morceau suivant }
        RePlay := j; { renvoie le morceau à jouer }
end;
{$LongGlobals-}
begin
        TLStartUp;      { initialisation du Tool Locator }
        i := MMStartUp; { initialisation du Memory Manager }
        WriteLn;        { affichage des messages }
        WriteLn(' Exemple d''utilisation de MJuke à partir
de TML PASCAL 1.00D');
        WriteLn;
        WriteLn(' (c) 1988  Stephan  HADINGER');
        WriteLn;
        WriteLn(' Musiques de  Stephane  BONAZZA');
        WriteLn('   - Dressing Room Piano.');
        WriteLn('   - Libre Attitude.');
        WriteLn('   - Desecration.');
        WriteLn;
        LoadOneTool(69, $101); { chargement de MJuke }
        MJStartUp(i, 0); { initialisation de MJuke }
        MJMount('1/MUS1.MJK', 1); { chargement des morceaux
}
        MJMount('1/MUS2.MJK', 1); { à partir du sous-
                                    directory }
        MJMount('1/MUS3.MJK', 1); { de l'application. }
{ gestion personnalisée de l'enchainement des morceaux }
{ cet appel n'est pas nécessaire }
        MJNext(@RePlay);
        MJJuke(1);          { mise en route du juke-box }
        WriteLn(' Appuyez sur "RETURN" pour quitter.');
        Readln;
        MJShutDown;         { fermeture de MJuke }
        MMShutDown(i);      { fermeture du Memory Manager }
        TLShutDown;         { fermeture du Tool Locator }

end.
```